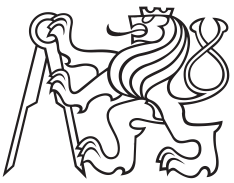
Master's thesis



Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

Design of Motion Primitives for a Hexapod Walking Robot Operating in a Rough Environment

Jakub Mrva
Robotics

May 2014
Supervisor: Ing. Petr Vaněk

Acknowledgement / Declaration

I would like to thank my advisor, Ing. Petr Vaněk, for being helpful whenever it was necessary, and I would also like to thank my parents for their continued support.

I hereby declare that this thesis is the result of my own work and all the sources I used are in the list of references, in accordance with the Methodological Instructions on Ethical Principles in the Preparation of University Theses.

Prague, May 12, 2014

.....
Jakub Mrva

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 12. 5. 2014

.....
Jakub Mrva

Abstrakt / Abstract

Kráčející roboti mají vzhledem ke své pohyblivosti velký potenciál pro použití v nerovném prostředí. Spolu s nejrůznějšími interními i externími senzory jako dálkoměrem, dotykovými senzory, kamerou, atd. se může pohybovat autonomně a vysoce efektivně.

Na rozdíl od jiných přístupů je v této práci použit sériově vyráběný šestinohý robot bez dalších přídavných senzorů. Jediným zdrojem informací o prostředí jsou jeho vlastní servomotory.

Pro tohoto robota je vyvinut takový vzor chůze, který mu umožní překonávat nerovný terén jen na základě vnitřních senzorů a za stálé podpory nejméně pěti nohou.

Klíčová slova: plánování pohybu; šestinohý kráčející robot; nerovné prostředí.

Překlad titulu: Návrh pohybových primitiv pro šestinohý kráčející robot pohybující se v nerovném prostředí

The crawlers have great potential to be successful in a rough environment due to their high motion capabilities. Along with other internal or external sensors like range-finder, force sensitive sensors, camera, etc., the robot can become an autonomous agent with very high efficiency.

In contrast with other approaches, a mass produced hexapod robot without any additional sensors is used in this thesis. The only information about the environment comes from its own actuators.

A custom gait which allows the robot traversing rough terrain based on the internal sensors only and with continuous support of at least five legs is developed for this robot.

Keywords: motion planning; hexapod walking robot; legged locomotion; rough environment.

Contents /

Tables / Figures

Chapter 1
Introduction

Although new technological principles are more and more amazing, they're often looking back to Mother Nature for inspiration. New technologies may be faster, robust, more powerful, more precise, but the natural optimization processes evolved during thousands of years are irreplaceable in our age where everything has to be done as fast as possible.

Nice example can be seen in robotics where legged locomotion is a rapidly growing domain. No animal has wheels, either.

1.1 Motivation

Various robots have been developed to perform various tasks. Considering only non-aerial vehicles, wheeled robots are good on flat terrains like pavements or roads (or Mars). Caterpillar tracks adds more flexibility to wheeled robots for traversing rough terrain. But the best results can be achieved with legged animal-like robots (crawlers). Their high degree of freedom (DOF), which is usually 3 DOF per leg, along with long leg range creates a vast domain of gaits and motion.

The crawlers have great potential in rescue operations, unstructured terrain exploring, load carriage and a lot of more, still undiscovered areas.

Robots, in general, are agents which deliberate regarding to the information they have about the outer environment. This includes a laser range-finder data of distances to the obstacles around the robot, 3-D camera images, pressure sensor data, etc. All of these data helps the robot to improve it's behavior.

But, when the robot doesn't have such sensors, the operating options are very limited. However, there's always some data available that can be utilized. Such data can come from actuators itself. Imagining a blind man focusing only on his sense of touch, a robot might be able to perform similarly.

1.2 Thesis Organization

This thesis is divided into several parts.

A brief summary of related existing work is written in Chapter 2. The hexapod that is used in this thesis is described in Chapter 3 including the analysis of the communication (Section 3.3) that can be achieved between the hexapod and the PC and the analysis of data reading described in Section 3.4.

A single leg description (Chapter 4) focuses on the compliance and stiffness of a leg (Section 4.1), on the direct and inverse kinematics (Section 4.2) and on the leg capability of perceive it's surroundings (Section 4.5).

The key part of this thesis – the newly developed gait – is presented in Chapter 5. Individual parts as the ground detection mechanism (Section 5.3), the body leveling transformation (Section 5.4) and the analysis of the body motion (Section 5.5) are described in detail.

The results of the testing of the gait performance are presented in Chapter 6.

Various related ideas and topics are discussed in Chapter 7.

Chapter 2
Related Work

Many research groups are working on a similar topic. However, their approaches can be quite different with different resources available.

2.1 Using Additional Sensors

There's a lot of crawlers with external sensors, but the one built by Boston Dynamics[1]) called BigDog[2]) [1] exceeds all of them. BigDog is about 1 m tall and long, weighs over 100 kg, is powered with a combustion engine and has over 50 sensors. BigDog can run at 7 km/h, carry over 150 kg and climb slopes up to 35°. He can also traverse rocky, muddy or snow-covered inclined surfaces which makes him the most advanced rough-terrain robot on Earth.

Another quadruped robot manufactured by Boston Dynamics called LittleDog[3]) also benefits from various sensor information. He's a bit smaller (0.3 m long, 0.18 m wide, 0.26 m tall, and weighs approximately 2.5 kg). In a solution [2] of a DARPA[4]) Learning Locomotion program, there is used a combination of a terrain scan from an external motion capture system with onboard inertial and force sensors to create a smooth and fast trajectory over optimized footsteps. LittleDog is strong enough to use his body dynamics to even jump on small obstacles.

A lot of robots wear a force sensitive resistor (FSR) with more or less DOF on each foot for better reading the ground reaction force. The force information can be used to classify the terrain beneath [3–5], or just to detect the ground and improve current gait performance [2, 6–10].

2.2 Internal Sensors Only

Another approach is to use only passive compliance for traversing rough terrain. Bio-inspired robots belong here, such as centipede robot [11] with flexible body which helps traversing obstacles, or fast running robots with underactuated legs [12] which benefits from a passive DOF that helps to store and distribute energy more effectively.

A wheel can also be designed to traverse obstacles smoothly, like an eccentric wheel [13]. Rotating legs are mounted on RHex[5]) (again by Boston Dynamics) which bounding and stair traversing capabilities were studied in [14]. Modifying the RHex legs can enable amphibious operations [15]. Combination of wheel and leg can lead to a wheeled-leg robot [16] which takes the positive effects from both wheel and leg approaches like high efficiency of motion on flat and high maneuverability on a rough terrain.

[1]) `http://www.bostondynamics.com/`
[2]) `http://www.bostondynamics.com/robot_bigdog.html`
[3]) `http://www.bostondynamics.com/robot_littledog.html`
[4]) `http://www.darpa.mil/`
[5]) `http://www.bostondynamics.com/robot_rhex.html`

The passive compliance and internal actuator information can also be used to classify the terrain [17].

The most related work is a research done by Palankar et al. [18–19]. They also worked on a technically blind hexapod robot with 18 DOF. But, in contrast with this thesis, they inserted a passive compliant actuator in each leg in order to measure the ground reaction force through it's displacement. Thus, their robot had a total of 24 DOF (18 active, 6 passive).

2.3 Other Approaches

Sensor data is not the only way to increase the rough terrain movement performance. Optimal foothold selection is also an important aspect. A planner is used to optimize the foothold selection online in the case of LittleDog robot [2]. The continuous follow-the-leader gait as is described in [20] reduces the demand on foothold selection by reusing them with the other legs. The footsteps can be replanned [21] if a fault is detected.

A stability criterion can be defined in order to increase the body stability over a rough terrain [20, 22].

Chapter 3
Hexapod Structure

This thesis is based on a mass produced hexapod robot *PhantomX Hexapod Mark II* built by Trossen Robotics[1]). The hexapod is shown in Figure 3.1 where we can see that it has six identical legs each with three identical actuators. The body hides a motherboard with an Arduino-compatible ArbotiX Robocontroller[2]) and a wireless XBee module[3]). An optional battery can be attached under the hexapod body instead of a cable wiring to an 11V adapter.

Figure 3.1. PhantomX Hexapod Mark II. (Taken from [23])

3.1 Dimensions

The dimensions of a default hexapod position are shown in Figure 3.2 in which the legs are in a symmetric default positions with the distances (in mm) as in the picture. The body is simplified, but each leg coxa joint (θ_C) is in the right position. All joints will be described later in Chapter 4.

The blue area represents the theoretical middle leg operating space in the ground projection plane (about 100 mm below the robot in default). The dashed arc represents the extended operating space under the robot. In practice, this is not achievable since the leg endpoint is following the stride line along the walking direction (gray line). Furthermore, a typical leg operating angle is much smaller in order to avoid leg collisions. The leg stride length is therefore limited too.

[1]) `http://www.trossenrobotics.com/`
[2]) `http://www.trossenrobotics.com/p/arbotix-robot-controller.aspx`
[3]) `http://www.digi.com/xbee/`

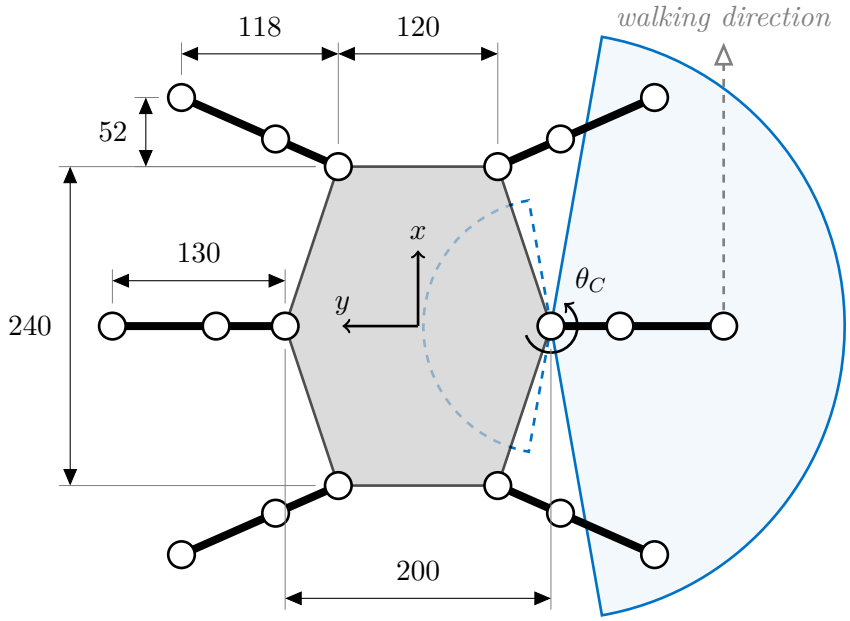


Figure 3.2. Schema of a default position of a hexapod. The leg positions are symmetric with the distances (in mm) as shown in the picture. The blue area represents the theoretical middle leg operating space in the ground projection plane (which is 100 mm below the robot in default). The blue dashed arc shows possible operating space assuming the leg can go under the robot (negative radius).

Figure 3.3. Dynamixel Servo AX-18A. (Taken from [24])

3.2 Actuators

Each leg has three joints which are actuated by an AX-18A Dynamixel servo shown in Figure 3.3.

The servo can be used in either wheel mode or joint mode. It has an operating angle of 300° in the joint mode. The servo resolution is 1024 units in the 300° operating radius, thus 1 unit equals 0.29°. Other parameters are given in the datasheet in Appendix D.

The servo has various read/write fields that can be set as they're well described in the manual [24]. We will mostly stay with the default initial values. Only compliance margin and slope parameters are optimized as will be described later in Section 4.1.

The servo has an internal position controller that runs at frequency 30 Hz and holds the last set position. The maximal and minimal position limits are handled in the running program on PC in the way that the position is never set beyond some limits to avoid mechanical damage on servos.

The controller is only proportional, so when compliance is set high enough and force is applied on a servo, it's position deviates. This effect is used in Section 4.5.

3.3 Communication

The communication runs at two stages. First between an external computer and main board and second between main board and all of the servos. The main computation runs on an ordinary external computer. New computed servo positions are then sent to the hexapod every 33 ms. An example of such time frame is shown in Figure 3.4.

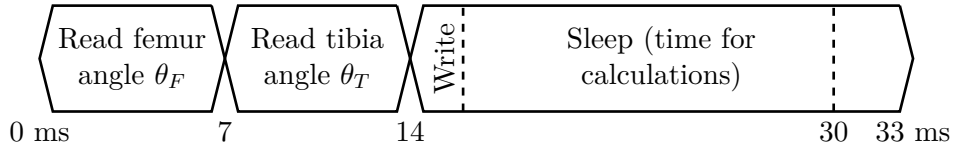


Figure 3.4. An example of a hexapod time frame.

The main program loop runs at frequency 33 Hz, hence 33 ms time frame. Each data reading lasts about 7 ms. In this example, the femur and tibia angle of a single leg (the moving one) is read in each cycle. Calculations are not very time consuming (only straightforward equations with the most recent leg positions), so new positions can be computed and sent to the hexapod quite immediately after the 14ms mark. Furthermore, the writing command doesn't block the running thread, so another computation can be made just after the data is sent until the 33ms mark.

It was experimentally tested that the bus on the hexapod is occupied for about 16 ms after receiving new positions. Any further reading attempt will be delayed for this amount of time, so only two data readings can be done during one frame. The rest time is filled with waiting for the next frame.

3.3.1 PC – Board

The transmission between an external computer and a hexapod can be done using either a standard USB cable or through a wireless XBee module. Since the wireless communication is neither as reliable nor as fast as the wired one, the wired connection si preferred.

It was experimentally tested that a wireless connection achieving the same response as in the example drawn in Figure 3.4 would need a time frame of about 100 ms, which is three times greater.

3.3.2 Board – Servo

The data frame given from an external computer contains desired positions of all of the 18 servos. The data is then sent to corresponding servos synchronously, so the servos will start moving to the new positions all at the same time.

In contrast, reading from servos works asynchronously, so the data can only be read individually.

3.4 Data Reading

As can be seen in the manual [24], there are a lot of fields that can be read. But since the robot is blind and the data reading is limited, we need to chose carefully the right ones. Besides the static fields like voltage limit, temperature limit, baud rate or compliance settings, the dynamic ones keeps the current values of e.g. moving speed, position, applied load, temperature or voltage. Since reading a single value of a single

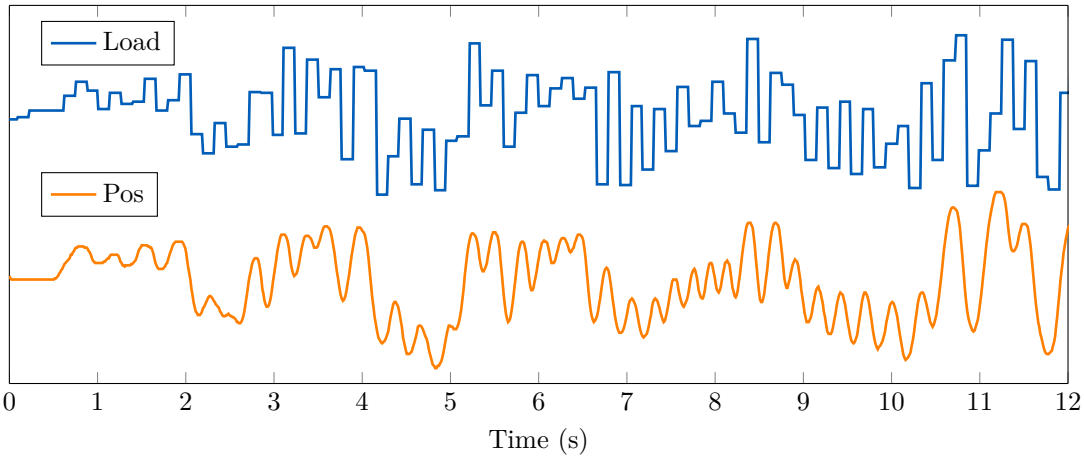


Figure 3.5. Comparison between reading load and position of a servo. Both data belong to the same servo and were obtained during the same motion (random shaking with a leg). Both data also have the same number of values. The main difference can be seen in the smoothness. The same load value is read multiple times in contrast with the position value which smoothly follows the harmonic wave.

servo is very time consuming and as was said above in Section 3.3 that we can read only two values during a time frame, the only relevant fields that make sense to read and that could be helpful are the current position and applied load. No other field can reflect the environment around hexapod.

These two quantities were read in a loop during an experiment shown in Figure 3.5. They both reflects the force applied on the servo. But the resolution is very different. Reading the applied load (which is inferred from the internal torque value; there's no torque sensor) shows the fact that this value is not refreshed too often, in contrast with reading the current position.

The position reading smoothly follows the harmonic wave created by hand-shaking with the leg. Therefore, it's the only value that will be used and read in this work. It's effects will be discussed later in Section 4.5.

3.5 Gaits

PhantomX hexapod robot contains several pre-programmed gaits. All of the gaits are based on a movement on a flat terrain. Although the robot can traverse some simple obstacles, he is blind and he assumes that there's a 100% flat ground beneath.

First, let us unify some frequently used terms. The following definitions 3.1 to 3.7 are adapted from [20].

Definition 3.1. The *transfer phase* of a leg is the period in which the foot is not on the ground. The leg state of a leg in transfer phase is 1.

Definition 3.2. The *support phase* of a leg is the period in which the foot is on the ground. The leg state of a leg in support phase is 0.

Definition 3.3. The *cycle time*, T, is the time for a complete cycle of a leg locomotion of a periodic gait.

Definition 3.4. The *duty factor*, β_i, is the time fraction of a cycle time in which leg i is in the support phase.

Definition 3.5. The leg *phase*, ϕ_i, is the fraction of a cycle period by which the contact of leg i on the ground lags behind the contact of leg 1.

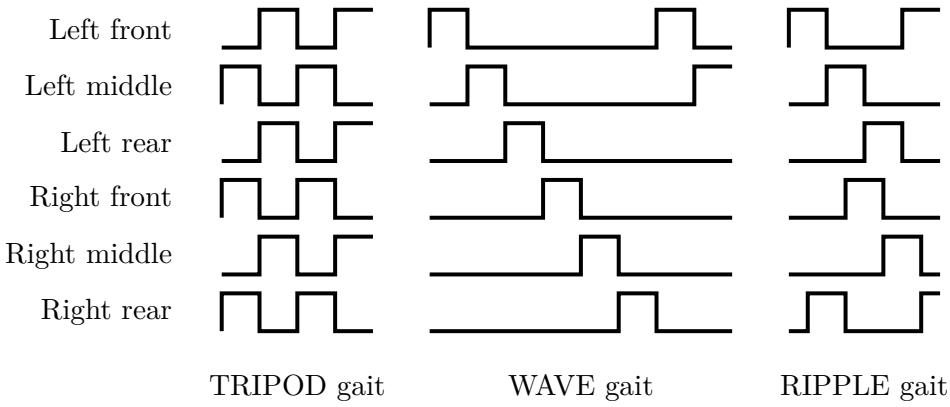


Figure 3.6. Timing plot for different gaits shows leg phasing and the transfer and support phases for all legs. (Adapted from [18])

Definition 3.6. The leg *stride*, λ, is the distance that the center of gravity translates during one complete locomotion cycle.
Definition 3.7. The leg *stroke*, S, is the distance through which the foot is translated relative to the body during the support phase.
Definition 3.8. The xyz-coordinate system has it's x-vector pointing in the forward walking direction, it's y-vector pointing perpendicularly to the left and it's z-vector pointing vertically upwards. They form a right-handed base.

The main gaits are shown in Figure 3.6. All legs share the same pattern of repeating transfer and support phases in each gait. The difference is in their phase shift ϕ_i which is closely related to the leg order. The difference between these gaits is mostly in the number of legs in the transfer phase at a time. The wave gait has always only one leg in the transfer phase. The ripple gait has always two legs in different periods of the transfer phase. The tripod gait has two groups of legs always in the opposite phases.

The leg order and determining the transfer and support phases influences greatly both the speed and the stability of the resulting gait. Because the stability is crucial in the rough terrain locomotion, we chose the ripple gait leg order with an extended support phase such that there's only one leg in the transfer phase at a time for our hexapod. This gait will be described in detail in Chapter 5.

3.5.1 Gait Controller

The gait controller handles the start and end positions in each step of a leg cycle. An example of these positions in a default triangular stroke is shown in Figure 3.7. Another example within a leg working space can be seen in Figure 4.4 later in Section 4.3.

A default ripple gait (as presented in Figure 3.6) is analyzed in Figure 3.7. The steps in a locomotion cycle are numbered from 1 to 6. It can be seen that steps 1 and 2 form the transfer phase as the leg is not on the ground during these two steps. The other four steps form the support phase. The duty factor is the ratio between the number of support phase steps and the total number of steps, hence $\beta = 2/3$.

The blue path represents the leg path with respect to the body, while the red path represents the same path with respect to the ground. The stride length can be measured from the red path since the points 2 to 6 share the same coordinates and form the footholds. The distance between them in two successive leg cycles gives us the leg stride λ.

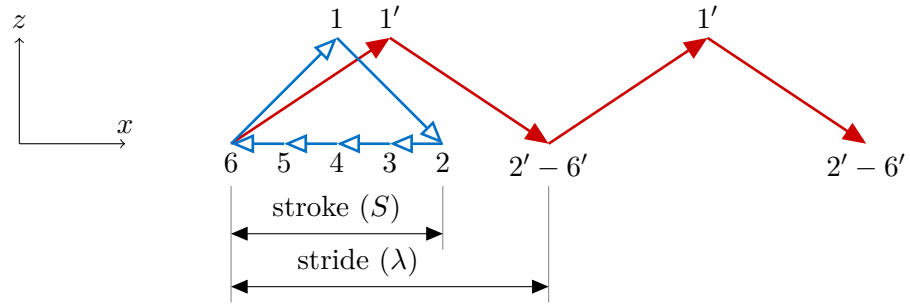


Figure 3.7. An example of a leg trajectory in body coordinates (blue) and ground coordinates (red). The duty factor is there $\beta = 2/3$ and stroke $S = \beta\lambda$.

The leg stroke can be measured from the blue path. It's the distance traveled by the leg during the support phase. Since the distance traveled by the hexapod is the same in each step, the stroke $S = \beta\lambda$.

Chapter 4
Hexapod Leg

A hexapod leg has three revolving joints and links between them. The exact measures (in mm) along with the overal leg schema are shown in Figure 4.1.

The first joint (Coxa – θ_C) has a vertical axis of rotation. The next adjacent joint (Femur – θ_F) has a horizontal axis as well as the last joint (Tibia – θ_T). The joints are named after the adjacent links.

The actuators are just in the position of the joints which increases the weight of a leg (unlike a pantograph leg). On the other hand, it increases it's maneuverability by adding another DOF.

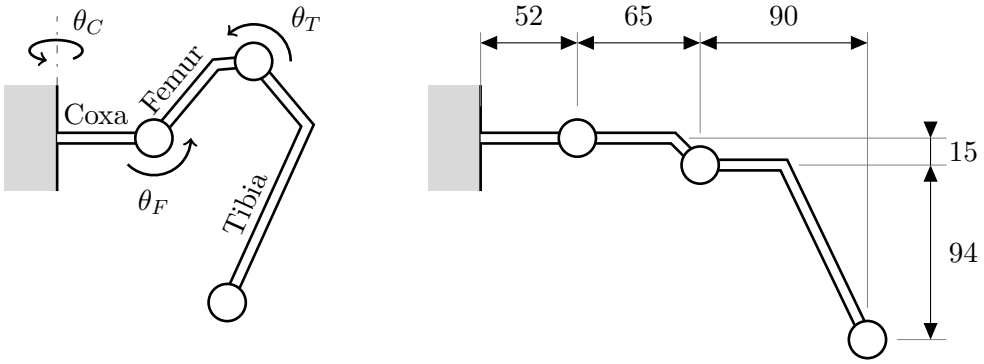


Figure 4.1. The Side view of a model of a hexapod leg. The Left picture shows a typical leg operating position while the right picture shows a leg with zero femur and tibia joint angles. The links are simplified, but the distances (in mm) shown in the right picture are those of the real hexapod.

4.1 Compliance

Compliance can be utilized well in passive joints to store energy especially in the case of running robots [12]. It has another importance for slowly walking robots in unstructured terrain. In this case, the load can be distributed better among all legs when they're not stiff.

The key factor of the success of the developed gait described later in Chapter 5 is the leg compliance setting. The legs (joints) have to be compliant enough to achieve a better resolution of leg displacement caused by an external (ground reaction) force (like in haptic sensing with compliant joints [25]).

As was mentioned earlier in Section 3.2, the actuators can be set with a desired compliance margin and slope. The influence of each parameter is illustrated in Figure 4.2. The level of output torque can be modeled using five parameters. Different behavior can be achieved in clockwise (CW) and counter-clockwise (CCW) direction. A minimal torque value (punch) can be set as well as a minimal position error needed to apply the torque for correcting the error.

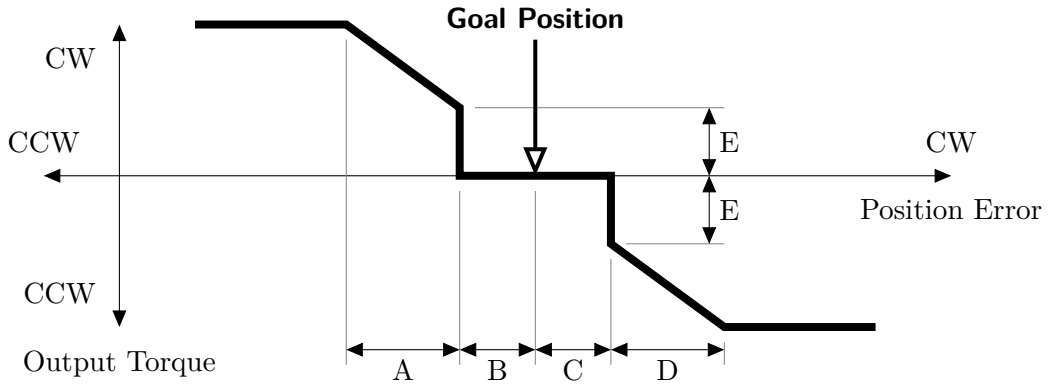


Figure 4.2. Compliance of the actuator AX-18A can be adjusted using five parameters (A–E). Both margin (C, resp. B) and slope (D, resp. A) can be set in clockwise (CW), resp. counter-clockwise (CCW) direction. Punch can be set using parameter E. (Adapted from [24])

Joint	Margin	Slope	Punch	(note)
Coxa	1	32	32	(default initial values)
Femur	0	128	32	(maximum slope value)
Tibia	1	64	32	(smallest w/o resonance)

Table 4.1. Compliance setting of all servos in both CW and CCW directions (see Figure 4.2). The margin is set in servo units. The slope has unidentified units, but the higher the value, the more flexible the servo is (see the documentation [24]).

The final values used in experiments are shown in Table 4.1. The same values are set for all servos and in both CW and CCW direction. The punch value is kept at the default initial value. This value is also a minimum value. The coxa joint keeps it's initial values as this joint has minimal influence on the overal gait performance.

According to the servo documentation [24], the slope value has no units, but the higher the compliance slope value, the more flexible the joint is. Actually, the values can vary from 2 to 128 (but only multiples of 2 are used internally; rounded down).

Assuming that the default minimum punch value is only a compensation to the servo nonlinearities (deadzone), the femur joint compliance setting tends to be a linear mapping of the position error to the torque value. Therefore a zero margin is set. The compliance slope is set to it's maximum value for the femur joint for a better resolution in the position error values.

The tibia joint is set stiffer than the femur joint because it doesn't reflect the ground reaction force due to it's position usually above the foot, anyway. Therefore, it's compliance is set to as small applicable values as possible without any resonance occurrence.

4.1.1 Resonance

When a joint is set too stiff, a resonance may appear. This happens especially in the case of the tibia joint because it's the last joint in the leg and the weight it carries is therefore very small.

Although the compliance slope value can be set up to 2, even a value of 16 can make the coxa joint oscillating slightly. The same happens when decreasing the initial margin from 1 to zero. Thus, it's recommended to left the initial compliance settings of the coxa joints.

4.2 Kinematics

Mapping between joint space and working space is quite nonlinear for a robot with revolving joints. Fortunately, a hexapod leg has only three joints and two of them have parallel rotation axes. This simplifies the kinematic task a lot.

Initially, the hexapod needed only mapping from working space into joint space (inverse kinematics) to make the legs move along a straight line instead of a circular one as would be more natural for revolving joints. But as will be described later in Chapter 5, it needs to be known where a leg currently stopped when it reached the ground. This involves the direct kinematics.

4.2.1 Coordinate Systems

According to the Denavit-Hartenberg notation [26] mentioned below, five Cartesian coordinate systems are used here to cover the path from the body to the foot. Note that different leg gives different coordinate systems (except the base one). The leg-relative ones are described in Figure 4.3.

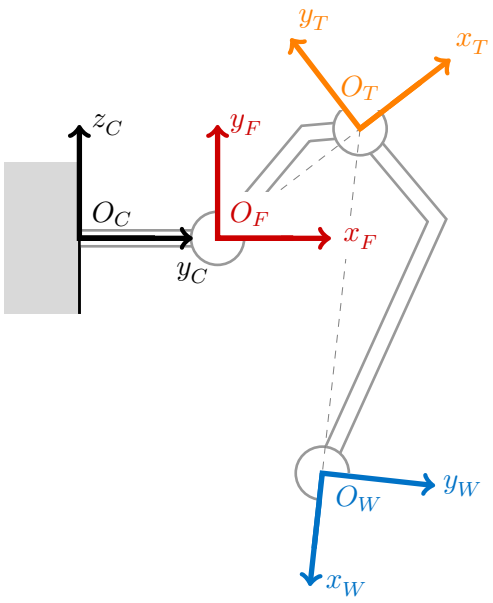


Figure 4.3. Defined coordinate systems of a hexapod leg that follows the Denavit-Hartenberg notation.

Definition 4.1. Coordinate system O_B lies in the COG with the x-axis heading forward and the z-axis heading upwards.
Definition 4.2. Coordinate system O_C is created by translating O_B into the rotation axis of the leg's coxa joint. No rotation is applied.
Definition 4.3. Coordinate system O_F lies in the femur joint with the z-axis lying in the femur rotation axis. The coxa joint lies on the negative part of the x-axis of O_F.
Definition 4.4. Coordinate system O_T lies in the tibia joint with the z-axis lying in the tibia rotation axis. The femur joint lies on the negative part of the x-axis of O_T.
Definition 4.5. Coordinate system O_W lies in the foot world coordinates.

The rotation of the coordinate system O_W doesn't need to be specified because nothing but it's origin will be used in the next sections. Thus, it could be arbitrary.

4.2.2 Direct Kinematics

Computing the direct kinematics of the hexapod, i.e., computing the position of the foot from the three given joint angles, is very straightforward. Assuming that the base coordinate system is translated from the center of gravity (COG) to the coxa joint rotation axis, the parameters given in Table 4.2 in accordance with the Denavit-Hartenberg notation are enough to compute the foot coordinates of the left middle leg of a hexapod.

i	α_i (rad)	a_i (mm)	θ_i (rad)	θ_i^{off} (rad)	d_i (mm)
C (coxa)	$\pi/2$	52	θ_C	$\pi/2$	0
F (femur)	0	66	θ_F	-0.22	0
T (tibia)	0	130	θ_T	-0.59	0

Table 4.2. A Denavit-Hartenberg notation of the left middle leg of a hexapod. Assuming that the z-axis of the base coordinate system lies in the rotation axis of the coxa joint.

The transformation which maps the joint angles $[\theta_C, \theta_F, \theta_T]$ to the world coordinates $[x, y, z]$ is inferred in Appendix E.1.

4.2.3 Inverse Kinematics

Although it would be more precise and model error tolerant when using an inverse kinematics of a floating base system [27–30], it's much more complicated and thus, we will stay with a fixed base.

As was mentioned above, the inverse kinematics of a hexapod leg is simplified using orthogonal and parallel axes. The resulting joint angles $[\theta_C, \theta_F, \theta_T]$ can be computed from the world coordinates of a foot $[x, y, z]$ using equations inferred in Appendix E.2.

4.3 Working Space

A leg working space is a mapping from joint space into working space and thus it's borders are determined by the joint angle limits. A projection of the working space of a middle leg in two different planes is shown in Figure 4.4.

The Left picture shows a side view of a hexapod with it's right middle leg. The blue highlighted area represents a projection of the working space into a plane parallel with the walking direction (x-axis). The plane is in the default distance from the coxa joint (which is 130 mm for a middle leg) of which the coordinate system (O_C) is drawn. The distance can be seen in the right picture where the same plane is projected as a gray dashed line. Ideally, when a default gait is applied, the foot coordinates of a middle leg never leaves this plane.

The right picture shows a rear view of a hexapod with the same leg (but in a different position). The blue, respectively orange arcs represents the foot position while moving the femur, respectively the tibia joint only and keeping the other joint fixed, so they form the operating space within an xy-plane of femur (or tibia or foot) coordinates system (see Section 4.2.1). The coordinate systems O_F, O_T and O_W rotates around the coxa joint rotation axis and follow the leg. The projection of the operating space in the right picture is therefore the same for arbitrary plausible rotation of the coxa joint.

The red path describes the same motion in both pictures – a default triangular move with leg stroke $S = 140$ mm (which is very high considering the possible leg collisions).

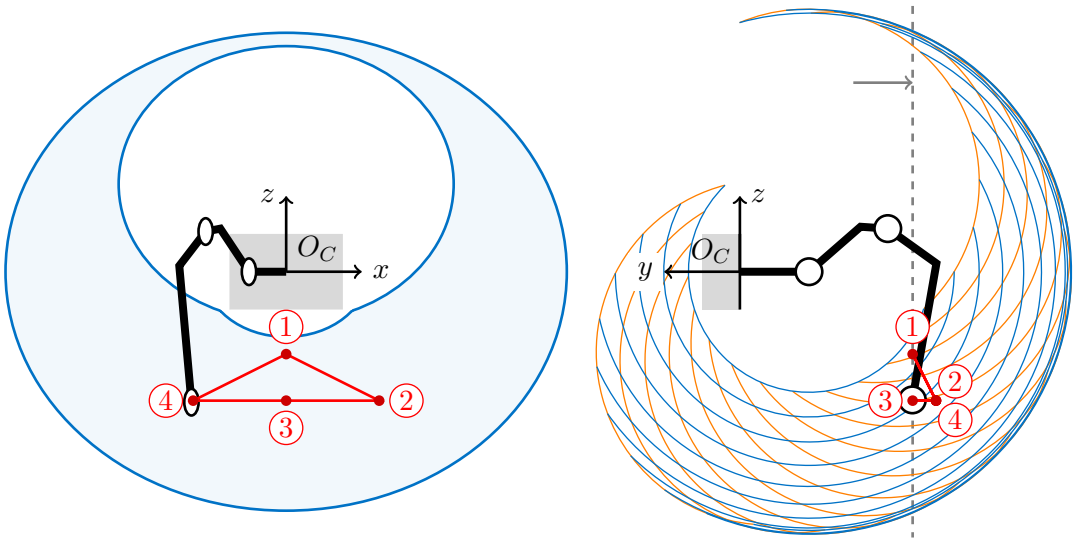


Figure 4.4. Working space of a hexapod leg. The left picture shows a side view of a hexapod with blue highlighted operating space of a middle leg within a plane in the default distance from a hexapod. The same plane is drawn in the right picture as a gray dashed line. The right picture shows the middle leg in the default position (point 3; in contrast with the left picture). The blue and orange arcs form represents the motion of femur and tibia joints and they form the operating space in a rotating plane around the coxa joint rotation axis. The red path shows a default triangular motion.

All of the points lie in the same plane in the hexapod coordinates (O_B or O_C), as can be seen in the left picture. But in the right picture, the path is projected to a plane (xy-plane of coordinate systems O_F, O_T and O_W) rotated around the coxa joint by a corresponding angle. Thus, the points are not on a line.

We can see that the working space of a leg has great reserves, especially in the y-axis projection, assuming that the red path shown in the pictures represents a motion that is large enough to need to take care of possible leg collisions.

4.4 Position Controller

The default triangular motion shown as a red path in Figure 4.4 is designed as a compromise between the hexapod height h above the ground and the distance between the legs and the body. The hexapod loses it's stability when the height h is bigger. On the other hand, the torque applied on servo is dependent on the leg lateral distance. The only critical moment is in the transfer phase in which the leg is moving up very close to the working space limits (see point 1).

This behavior (operating near the limits) is allowed when the hexapod is following a deterministic gait, i.e., the legs are following still the same path within the working space and the only disturbance can come from the outer environment. Such a gait can be designed solely for a flat ground operations which is not the case of this work. We need to achieve a higher fault tolerance for a new developed gait described in Chapter 5.

Firstly, the gait *lifting level* is very low (but high enough for a flat ground operations), so we need to increase this level to avoid obstacle hitting by a leg in the transfer phase.

Secondly, we need to increase the leg depression by adding a *drop level*. This allows the leg reach a terrain hole or a beginning of a slope down terrain.

Thirdly, the default lifting level is near it's limits. In order to increase the robustness and create a bigger reserve, we have to slope the path such that the foot position at the lifting level is farther from the body and the foot position at the drop level is closer to the body.

All of the three improvements are shown in Figure 4.5 on a path of a zero-stride gait (the hexapod is running in place). The terms *lifting level* and *drop level* are shown as horizontal lines here. The background image is a magnified cutout of Figure 4.4. The blue path represents the path between the points 1 and 3 in Figure 4.4 (assuming a zero stride, the path is a straight line now). The orange path represents the improved path following the three points of the proposal above. Because the inverse kinematics is computed for the initial and terminal points only, the points between are created by a linear interpolation of the computed joint coordinates of the outer points in the joint space. Thus, the resulting path is not a straight line in the working space.

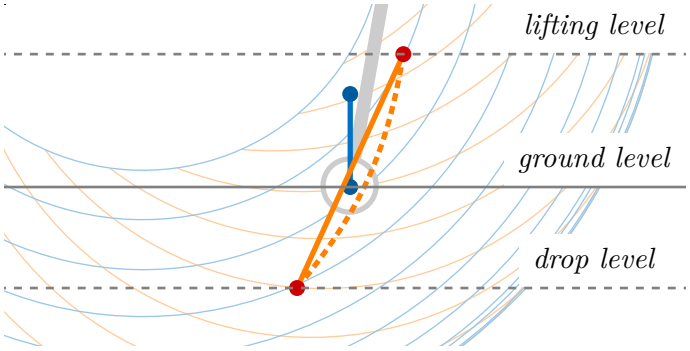


Figure 4.5. Comparison between a default (blue) and an improved (orange) leg path within a leg working space. The dashed orange path shows the actual path with respect to the interpolation in the joint space.

As was broached above, the leg motion is partitioned. A number of steps is computed from the given transition time divided by the length of a time frame (33 ms – see Section 3.3). Joint coordinates of a given terminal point are computed using the inverse kinematics. All steps are distributed uniformly along a line in the joint space connecting the terminal point joint coordinates and the current position joint coordinates.

4.5 Sensing

This is the key part of the gait's ability to successfully walk over uneven terrain. The interaction between the leg and the ground (the ground reaction force) needs to be measured somehow. Due to the lack of other sensors, we have to rely on the servo positioning system. As was mentioned in Section 3.2, the servo position controller (included in the servo itself) is only proportional. It's position deviates when a load is applied on the servo. This deviated position can be read and compared with the desired one. Using the direct kinematics, we can reconstruct the deviated foot position by reading the actual femur and tibia joint angles (assuming that the coxa joint isn't directly affected by the leg motion as it's rotation axis and the foot path lie in the same plane).

According to the data reading, we can calculate with three variables. The femur angle displacement, the tibia angle displacement and the computed foot position displacement. Each of the variables acts differently and has a different resolution.

The foot position displacement in the z-axis would be enough for estimating the ground reaction force if the path was a line perpendicular to the ground. But as can be seen from Figure 4.5, the path is inclined from a vertical direction.

In Figure 4.4 (or 4.5), we can see that the default leg operating position lies approximately under the tibia joint. The ground reaction force (acting upwards) therefore has only a little influence on the tibia angle displacement. Moreover, the leg motion path is sloped so it better traces the femur joint rotation (the blue arcs in the pictures) and thus the femur joint angle reflects the ground reaction force quite authentically.

Although the femur angle displacement mostly shows similar trends as the z-axis displacement, it has about twice as large resolution (comparing servo units vs. mm).

An example of usage of the sensing behavior in the ground detection case is shown in Figure 5.2 later in Section 5.3.

Chapter 5
Blind-Tread Gait

The robot has no external sensors. It has neither a camera nor a laser scanner. The only feedback from the outer world the robot has is the joint angle displacement of its servos. Thus, the only information comes from the contact with the ground.

The new developed gait presented here has to manage these facts. First of all, the robot is technically blind. So, the gait evolved in something like slow cautious tread, hence the name *Blind-Tread Gait*. When a leg is moving down, it can hit the ground in theoretically arbitrary moment. In order to obtain more precise results, only one leg at a time is moving. This avoids most of the interference between the legs. The time that the leg in the transfer phase spends on a moving-down step is not deterministic because the leg is moving down with a constant speed while the distance from the ground is unknown. Therefore, synchronizing more than one leg at a time is not an easy task. Moreover, according to the communication limits introduced in Section 3.3, only two servo position values (e.g. femur and tibia joint angles) can be read in one 33ms time frame, so there's no time left for reading the position of another leg.

In a related work of Palankar et al. [18–19], they used a similar approach with a blind hexapod robot. They measured the ground reaction force by a passively compliant joint inserted between the body and the coxa joint in the open kinematic chain. Each leg has it's own distributed feedback controller which controls the elevation or depression rate of a leg according to the measured displacement in the passive compliant joint in order to keep the leg supporting the robot (keep in touch with the ground). Their controller is therefore an upgrade to a default leg position controller and can be applied on arbitrary gait.

In this work, such an approach is not feasible since a lot of hardware upgrade would have had to be done. We have only one controller with limited capabilities commanding all six legs. Thus, our approach is to develop a brand new gait with a cycle divided into several steps which are statically stable. There are no dynamic transitions between the steps, so it's kind of a stop motion.

5.1 Overall View

An overall picture of the blind-tread gait is shown in Figure 5.1. This diagram shows a leg cycle of an active leg which is searching for a new foothold. This cycle is then repeated for the other legs and six consecutive cycles combine to a gait cycle. The legs are alternating in a ripple gait with a given order: LF – RR – LM – RF – LR – RM[1]).

At the start (and at the end) of each cycle, all legs lay on the ground and the body is leveled equally with respect to the average positions of all legs (see more below in Section 5.4.1). Then, one leg moves up and forward and begins to approach the ground. Sensor data is gathered along the leg path, as is described below in Section 5.2. Once the ground is detected (using rule described below in Section 5.3), the leg stops moving

[1]) LF = left-front, RR = right-rear, LM = left-middle, etc.

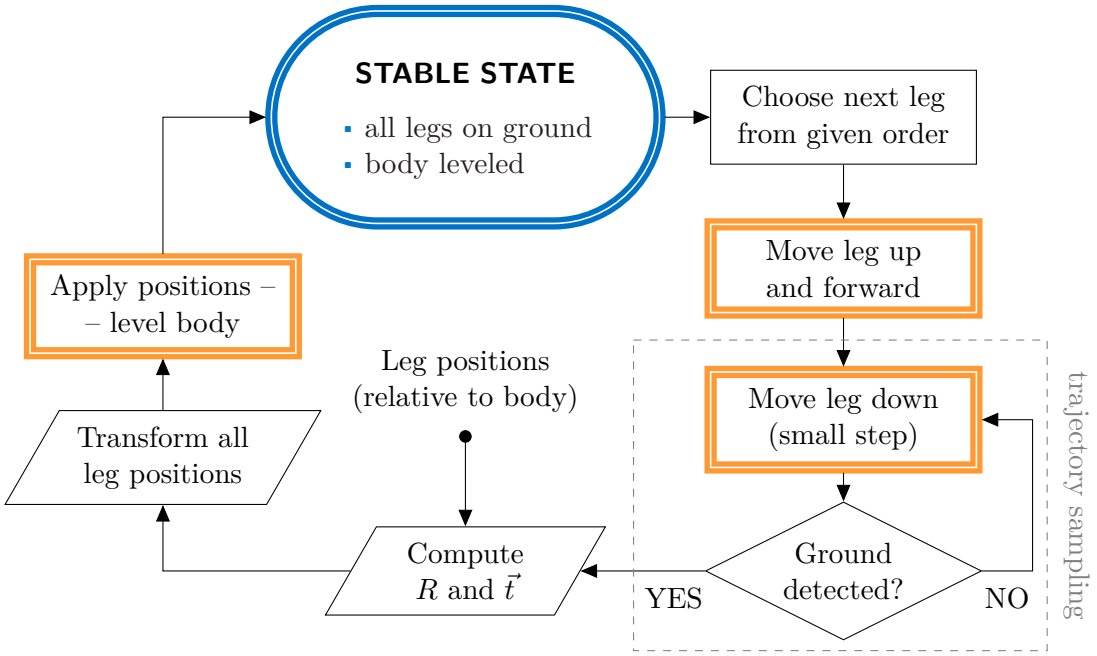


Figure 5.1. Blind-tread gait diagram. Only one leg at a time is moving except the level-body step where all legs are assigned new positions with respect to the body but the contact points (footholds) stay the same.

and memorizes its new position (with respect to the body – coordinate system O_B). All six leg positions are then used in a linear regression function (see more below in Section 5.4.1) which computes an approximation of a flat ground (plane) providing parameters a, b and c – the coefficients of the plane's equation. These parameters form a rotation matrix R which models the rotation of the regression plane with respect to O_B (derived in Section 5.4.3). Other parameters (t_x, t_y and t_z) derived in Section 5.4.4 ensure that the average of all new leg x-coordinates and y-coordinates will be zero and the average of z-coordinates (in the least square mean) will equal the default body height h of the robot. The body is then leveled to equal the angle and distance with the computed plane by applying the new leg positions.

The only steps in the blind-tread gait leg cycle shown in Figure 5.1 in which a motion occurs are highlighted in orange. The transfer phase consists of a leg moving up, forward and down. The support phase consists only of the body leveling in the six consecutive leg cycles.

The resulting body trajectory is studied in Section 5.5.

5.2 Trajectory Sampling

The leg depression trajectory towards the terminal point at the drop level is sampled into numerous interpolation steps (see Section 4.4). At the end of each interpolation step, a data reading is executed from which an information about the acting ground reaction force is acquired.

Thus, we obtain a discrete function of the ground reaction force depending on the foot position along the depression trajectory.

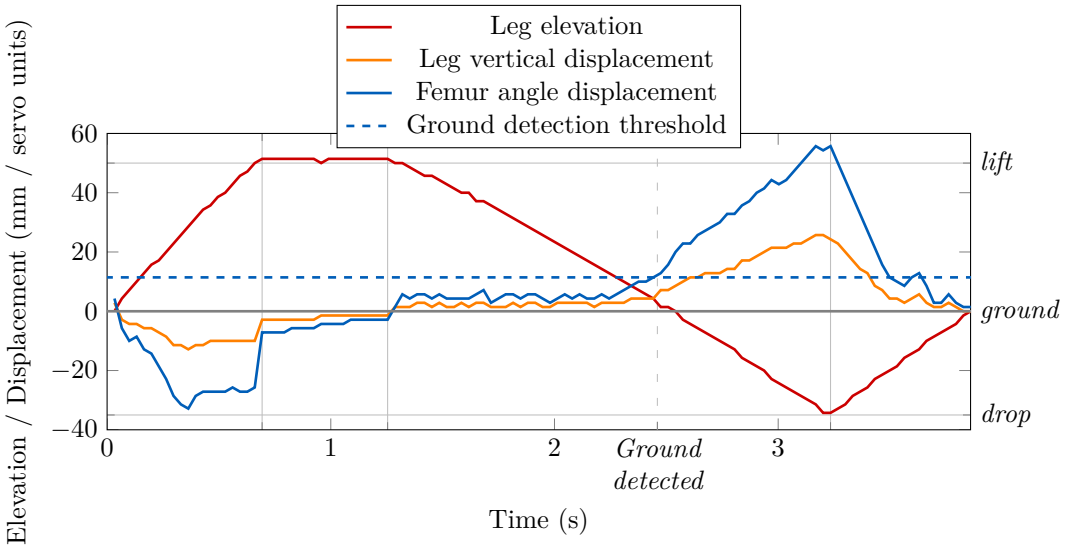


Figure 5.2. A complete active-leg cycle. Vertical lines divide the cycle into four steps: up, forward, down (with ground detection – dashed line) and leveling back. In this example, the down step is not interrupted when the ground is detected, but keeps the leg depressing to the drop level instead to show how the ground reaction force affects the overal leg position vertical displacement.

5.3 Ground Detection

Following the sensing behavior described in Section 4.5, we can estimate the moment when the leg actually hits the ground. An example of a complete leg cycle of an active leg (performed on a flat ground) is shown in Figure 5.2.

In the first step, the leg is elevating following the red graph. When the elevation transition time is set low and hence the elevation speed is high, the leg inertia holds the leg down and the femur joint deviates to it's negative direction. The femur servo compliance is set very high (see Table 4.1 in Section 4.1) and so it's angle displacement is high too.

The femur servo slowly recovers during the horizontal motion and due to the leg inertia, it begins to deviate in it's positive direction when depressing. The resulting angle displacement grows with a higher transition speed and can be accidentally mistaken for a ground-hitting event. So the depression speed has to be set carefully (as in Figure 5.2).

A ground hitting event is detected here but the motion doesn't stop to show the leg sensing capabilities.

The computed leg z-axis displacement represents a perpendicular ground reaction force which can be used in the meaning of a support level of a leg. The femur angle displacement has a similar trend and a bigger resolution, so solely this raw value is used for detecting the ground-hitting event instead of using z-axis displacement computed through the direct kinematics.

A threshold value corresponding to a support level of an approximately even force distribution among all of the legs have to be chosen carefully. The value is most affected by the weight of the robot.

When the ground is detected, the new foothold position is remembered.

5.4 Body Leveling

The next step of the blind-tread gait is the recalculation of the body position[1]) in the way that it better fits the newly achieved active-leg position. This needs to be done because when a hexapod is approaching a sloped terrain, the body would either hit the terrain or enter a position in which the terrain is unreachable if the body remains in the same position (height).

However, there can be found infinitely many body positions for given foot positions. The only limitation is that all of the foot positions have to be within the working space of corresponding leg. Each body position offers different possibilities of movement depending on how close to the working space limits the legs are.

Because the hexapod has to walk over a rough terrain without any perception about the terrain ahead, there has to exist an *equilibrium body position* somewhere „in the middle“ of the body positions space. Such a body position offers balanced possibilities of movement in all directions.

Since the body has no actuator itself, the foot positions has to be recalculated instead. Assuming that all legs lay on the ground, applying their new positions will enforce the body to move while the legs keep their footholds. It's obvious that the new foot positions have to keep the same distances between each other as the old ones. This is preserved when the change of the foot coordinates is induced by a motion that can be modeled by an orthonormal rotation matrix R or a translation vector $\vec{t}$ or both.

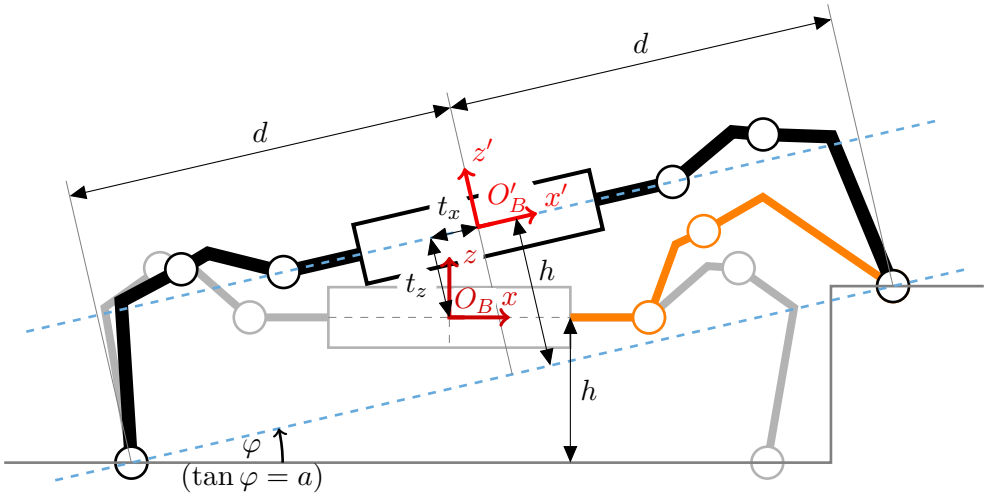


Figure 5.3. Demonstration of body leveling process. A new leg position (orange) enforce the body position to move from O_B to O'_B.

Figure 5.3 shows a schematic diagram of body leveling process. The schema is exaggerated (such a leg range is unreachable in real) and simplified (reduced number of legs from six to two) for better reading and clearer pointing out of the key principles.

The figure shows a hexapod in the default position (gray). Then, a leg (orange) has found it's new foothold on a higher terrain. A regression plane $z = ax + by + c$ (bottom blue dashed line) is computed. The plane goes through actual leg positions (with more legs, it goes through as close as possible in the sense of least square mean). The slope of the regression plane forms the rotation matrix R which rotates around the initial COG (origin of the coordinate system O_B). The rotated coordinate system (which is

[1]) The *body position* is meant in a 6D space containing both position and orientation.

not drawn here, but later in Figure 5.4) is then translated in it's z-axis by t_z to achieve the height h above the regression plane (the same as was above the ground before) and translated in it's x-axis by t_x so that the sum of the x-coordinates in the resulting coordinate system O'_B is zero. The translation in the y-axis by t_y is analogical to the x-axis. Thus, the body is kept „in the middle" of the legs and at a constant height h above the linear approximation of the ground.

5.4.1 Linear Regression

A linear and planar (and much more types of) fitting using the linear regression is nicely described for example in [31]. Although planar fitting using orthogonal regression would be more accurate (because this is what we tend to achieve – an average leg depression with respect to the tilted body), it is much more complicated and the resulting pitch and roll angles will be small enough to keep the difference between orthogonal and vertical method under resolution.

Given the foot positions in the initial coordinate system O_B, we need to determine the parameters a, b and c so that the plane $z = ax + by + c$ best fits the positions, e.i., their squared distance from the plane is minimized. The computation of the parameters is inferred in Appendix E.3.

5.4.2 Transformation

Given the foot positions $\{[x_i, y_i, z_i]\}_{i=1}^{6}$ in coordinate system O_B and the linear regression plane in the form $z = ax + by + c$ (also in O_B), we can construct a transformation matrix $A^{4\times4}$ in homogeneous coordinates which encapsulates both the rotation matrix $R^{3\times3}$ and the translation vector $\vec{t}$ (described in Figure 5.3).

The body transformation uses the *alibi* (active) representation of motion[1]). Unlike the classical transformation, we use a translation vector $\vec{t}$ expressed in the rotated coordinate system O'_B (so we use $R\vec{t}$ instead of $\vec{t}$). This will be helpful later in Section 5.4.4 when parameterizing this vector. The new body coordinates $[x'_B, y'_B, z'_B]$ can be therefore written as

$$\begin{bmatrix} x'_B \\ y'_B \\ z'_B \\ 1 \end{bmatrix} = \begin{bmatrix} & R & & R\vec{t} \\ & & & \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_B \\ y_B \\ z_B \\ 1 \end{bmatrix}. \tag{1}$$

Note that the leg positions remains unchanged in the *alibi* representation of motion in which the coordinates are computed with respect to the initial coordinate system O_B.

As was mentioned earlier, the actuators are not in the body but in the legs actually. The desired leg coordinates which enforce the body to move to it's coordinates expressed in Equation (1) has to be computed using the *alias* (passive) representation of motion. This involves a coordinate system attached to the moving body moving to it's new position O'_B. Computing the leg coordinates $[x'_i, y'_i, z'_i]$ in O'_B implies inverting the transformation matrix in the equation above. Since we are searching for an orthonormal rotation matrix R, it has a property $R^{-1} = R^T$ and $R^T R = I$.

[1]) The terms *alias* and *alibi* were introduced in the classical monograph [32]. All of the background mathematics used here is nicely described in [33–34].

The leg coordinates (using the inverse transformation inferred in Appendix E.4) can be computed as

$$\begin{bmatrix} x'_i \\ y'_i \\ z'_i \\ 1 \end{bmatrix} = \begin{bmatrix} & R^T & & -\vec{t} \\ & & & \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}. \tag{2}$$

There are two unknowns that have to be solved – the rotation matrix R and the translation vector $\vec{t}$. Everything else is straightforward.

5.4.3 Rotation

The rotation matrix R represents a motion of rotating the coordinate system O_B such that it's new coordinate axes vectors are the same as in the coordinate system O'_B but without respect to their origins (the translation will be handled later), so we obtain an auxiliary coordinate system O''_B (described in Figure 5.4). It has also a significant relationship with the regression plane.

Since we're rotating orthonormal vectors of coordinate system O_B, the rotation matrix vectors can be directly obtained from the vectors of coordinate system O''_B. If we expand the basis of the coordinate system O_B to it's basic vectors $\left[\vec{b}_x, \vec{b}_y, \vec{b}_z\right]$, we can express the basic vectors of O''_B similarly as in Equation (1) and we get

$$\left[\begin{array}{c|c|c} & & \\ \vec{b}''_x & \vec{b}''_y & \vec{b}''_z \\ & & \end{array}\right] = R' \left[\begin{array}{c|c|c} & & \\ \vec{b}_x & \vec{b}_y & \vec{b}_z \\ & & \end{array}\right], \tag{3}$$

where R' is an orthogonal rotation matrix whose basic vectors $\left[\vec{R}'_x, \vec{R}'_y, \vec{R}'_z\right]$ have the same directions as the basic vectors of R but they don't have the same magnitude.

We know that the basic vectors of O_B $\left[\vec{b}_x, \vec{b}_y, \vec{b}_z\right]$ form a standard basis. The columns of matrix R' are therefore

$$\left[\begin{array}{c|c|c} & & \\ \vec{b}''_x & \vec{b}''_y & \vec{b}''_z \\ & & \end{array}\right] = R' \left[\begin{array}{c|c|c} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{c|c|c} & & \\ \vec{R}'_x & \vec{R}'_y & \vec{R}'_z \\ & & \end{array}\right]. \tag{4}$$

Although we don't know the actual coordinates of axes of O''_B, we have several clues that should limit the infinite space of all possible orientations enough to get the only solution:

- The x- and y-coordinate axes are parallel to the regression plane $z = ax + by + c$.
- The x-axis is heading forward in the walking direction.
- The rotation matrix R' is orthogonal.

The second point ensures that the x-axis of O''_B lies in the xz-plane of both coordinate systems O_B and O'_B. This plane is drawn in Figure 5.4. Together with the first point, we can draw the x-axis in the xz-plane as x'' of coordinate system O''_B which represents the rotation only. The slope of x'' with respect to x is the same as the slope of the regression plane projected into the xz-plane and it can be computed directly from the plane's equation $z = ax + by + c$.

$$\frac{\partial z}{\partial x} = a = \tan\varphi, \tag{5}$$

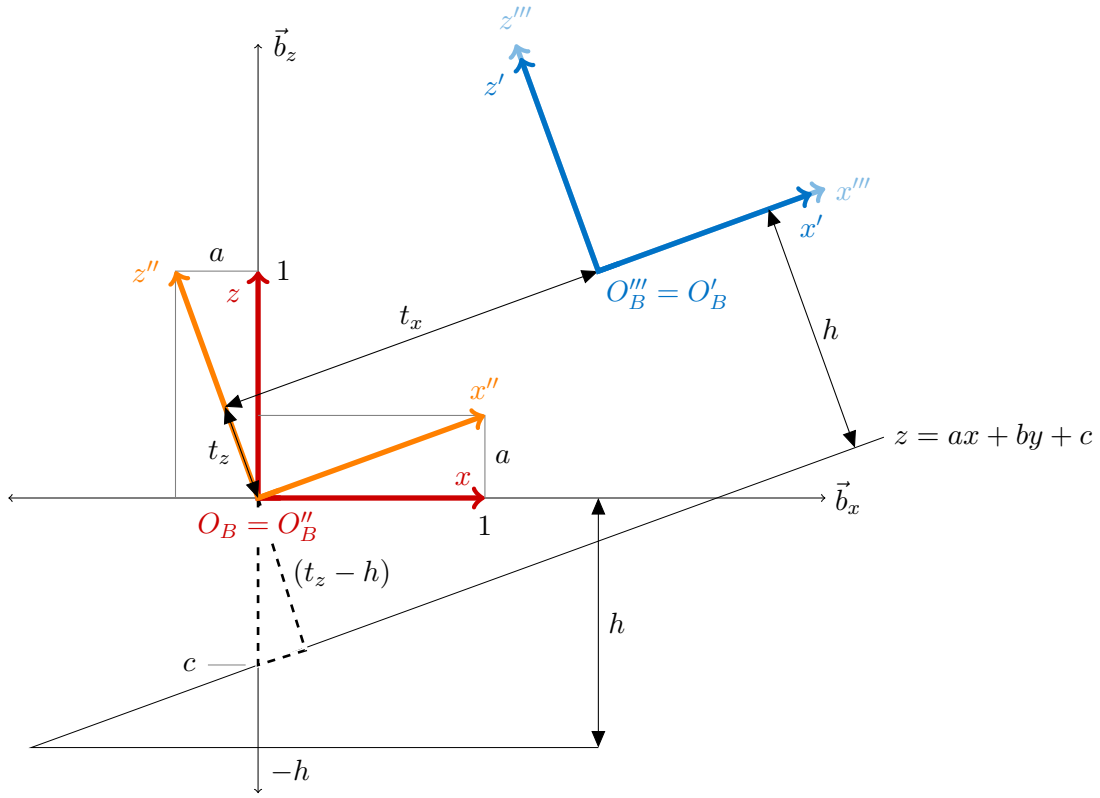


Figure 5.4. Transformation diagram of parameterization of the body leveling. The y-axes are omitted here. The transformation starts with the initial basis O_B (red). A rotated basis O''_B (orange) is found such that it's x-axis coordinates follows the tangent of the slope of the regression plane. A translation vector in O''_B is applied to find the origin of O'''_B and terminal basis O'_B. A normalization was applied to create an orthonormal basis O'_B from an orthogonal basis O'''_B.

where φ is the angle between x'' and x (see Figure 5.3). According to the tangent, the ratio between the z- and x-coordinates of $\vec{b}''_x$ therefore equals a.

The coordinates of the basic vector $\vec{b}''_x$ are then fully parameterized as

$$\vec{b}''_x = \begin{bmatrix} 1 \\ 0 \\ a \end{bmatrix} = \vec{R}'_x. \tag{6}$$

The first point from the list above also tells us that (assuming we have an orthogonal basis) the z-axis has to be perpendicular to the regression plane. Such a vector is a normal vector of the plane and it can be computed directly from the plane's equation which we rewrite into a more general form

$$\begin{aligned} z &= ax + by + c \\ 0 &= ax + by - z + c. \end{aligned} \tag{7}$$

The coordinates of the normal vector are the coefficients at the x, y and z variables in the general form of the plane's equation. The normal vector can be multiplied by an arbitrary scalar, so we will multiply it by -1 to get a vector heading upwards as is heading the z-axis vector of O_B. Hence we get the parameterized coordinates of the

basic vector $\vec{b}''_z$ as follows

$$\vec{b}''_z = \begin{bmatrix} -a \\ -b \\ 1 \end{bmatrix} = \vec{R}'_z. \tag{8}$$

We can now partially fill in Equation (4) by substituting the parameterized basic vectors from Equations (6) and (8) to get an almost fully parameterized rotation matrix

$$R' = \left[\begin{array}{c|c|c} 1 & & -a \\ 0 & \vec{b}''_y & -b \\ a & & 1 \end{array}\right]. \tag{9}$$

The rotation matrix R' is orthogonal, so the third basic vector has to be linearly independent of the two others, i.e., their scalar multiplication has to be zero, hence

$$\vec{b}''_x \vec{b}''^T_y = 0 = \vec{b}''_z \vec{b}''^T_y. \tag{10}$$

There are infinitely many of such vectors that forms an orthogonal basis. Let us define the elements of $\vec{b}''_y$ as $\vec{b}''_y = [y_1, y_2, y_3]^T$ and choose one of the vector's coordinates manually. Since we want to keep the notation consistent, we will follow Definition 3.8 and choose

$$y_2 = 1, \tag{11}$$

which ensures that the y-axis will be pointing to the left side of the hexapod perpendicularly to the walking direction. It is also a safety choice because the basic vector $\vec{b}''_y$ will always have coordinates close to $[0, 1, 0]^T$ in real operations[1]).

We can then formulate a set of equations using the scalar multiplication from Equation (10) as

$$\begin{aligned} \vec{b}''_x \vec{b}''^T_y &= \quad y_1 + \quad + a y_3 = 0 \\ \vec{b}''_z \vec{b}''^T_y &= -a y_1 - b y_2 + \; y_3 = 0. \end{aligned} \tag{12}$$

Expressing y_3 from the second equation and substituting it to the first equation (and substituting from Equation (11)), we can compute y_1 by following

$$\begin{aligned} y_3 &= a y_1 + b \\ \\ y_1 &= -a y_3 \\ y_1 &= -a(a y_1 + b) \\ y_1 &= -a^2 y_1 - ab \\ y_1 &= \frac{-ab}{a^2 + 1}. \end{aligned} \tag{13}$$

The last coordinate y_3 follows a simple computation

$$\begin{aligned} y_3 &= a y_1 + b \\ y_3 &= \frac{-a^2 b}{a^2 + 1} + \frac{b(a^2 + 1)}{a^2 + 1} \\ y_3 &= \frac{b}{a^2 + 1}. \end{aligned} \tag{14}$$

[1]) If we chose for example $y_3 = 1$, the other coordinates would be infinite, which should never happen.

The resulting basic vector $\vec{b}''_y$ is simplified by multiplying by (a^2+1) and hence

$$\vec{b}''_y = \begin{bmatrix} -ab \\ a^2+1 \\ b \end{bmatrix} = \vec{R}'_y. \tag{15}$$

In order to fulfill the orthonormality of the rotation matrix R, we need to normalize the three basic vectors $[\vec{b}''_x, \vec{b}''_y, \vec{b}''_z]$ computed in Equations (6), (15) and (8).

Let us define $||\vec{x}|| = \sqrt{\vec{x} \cdot \vec{x}}$ as the Euclidean norm of $\vec{x}$. We conclude that

$$\begin{aligned} ||\vec{b}''_x|| &= ||\vec{R}'_x|| = \sqrt{a^2+1} \\ ||\vec{b}''_y|| &= ||\vec{R}'_y|| = \sqrt{a^2b^2+a^4+2a^2+1+b^2} \\ ||\vec{b}''_z|| &= ||\vec{R}'_z|| = \sqrt{a^2+b^2+1}. \end{aligned} \tag{16}$$

Dividing each basic vector by it's norm, we obtain an orthogonal basis with unit vectors – an orthonormal basis. After switching the notation by using Equation (4) where $\vec{b}''_i = \vec{R}'_i$, we get an orthonormal rotation matrix

$$R = \left[\begin{array}{c|c|c} \frac{\vec{R}'_x}{||\vec{R}'_x||} & \frac{\vec{R}'_y}{||\vec{R}'_y||} & \frac{\vec{R}'_z}{||\vec{R}'_z||} \end{array}\right] = \left[\begin{array}{c|c|c} 1 & -ab & -a \\ 0 & a^2+1 & -b \\ a & b & 1 \end{array}\right] \begin{bmatrix} ||\vec{R}'_x|| & 0 & 0 \\ 0 & ||\vec{R}'_y|| & 0 \\ 0 & 0 & ||\vec{R}'_z|| \end{bmatrix}^{-1}. \tag{17}$$

5.4.4 Translation

The vector $\vec{t} = [t_x, t_y, t_z]$, as is partially described in Figure 5.4, represents the translation between the coordinate systems O_B and O'_B with it's coordinates written in the rotated basis of O'_B (or O''_B). This uncommon notation lets us write the transformation of the foot coordinates as in Equation (2) where the translation looks a lot simpler. Written in more detail, we obtain

$$\begin{bmatrix} x'_i \\ y'_i \\ z'_i \\ 1 \end{bmatrix} = \begin{bmatrix} & R^T & & -\vec{t} \\ & & & \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} = \begin{bmatrix} ---\vec{R}'^T_x --- / ||\vec{R}'_x|| --- & -t_x \\ ---\vec{R}'^T_y --- / ||\vec{R}'_y|| --- & -t_y \\ ---\vec{R}'^T_z --- / ||\vec{R}'_z|| --- & -t_z \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}. \tag{18}$$

When we rewrite the equation line by line, we get

$$\begin{aligned} x'_i &= \frac{\vec{R}'_x [\, x_i \quad y_i \quad z_i \,]}{||\vec{R}'_x||} - t_x \\ y'_i &= \frac{\vec{R}'_y [\, x_i \quad y_i \quad z_i \,]}{||\vec{R}'_y||} - t_y \\ z'_i &= \frac{\vec{R}'_z [\, x_i \quad y_i \quad z_i \,]}{||\vec{R}'_z||} - t_z \end{aligned} \tag{19}$$

We can then sum up all the legs in each equation and hence

$$\begin{aligned}
\sum_{i=1}^{6} x'_i &= \frac{\vec{R}'_x \left[\sum_{i=1}^{6} x_i \quad \sum_{i=1}^{6} y_i \quad \sum_{i=1}^{6} z_i \right]}{||\vec{R}'_x||} - 6t_x \\
\sum_{i=1}^{6} y'_i &= \frac{\vec{R}'_y \left[\sum_{i=1}^{6} x_i \quad \sum_{i=1}^{6} y_i \quad \sum_{i=1}^{6} z_i \right]}{||\vec{R}'_y||} - 6t_y \\
\sum_{i=1}^{6} z'_i &= \frac{\vec{R}'_z \left[\sum_{i=1}^{6} x_i \quad \sum_{i=1}^{6} y_i \quad \sum_{i=1}^{6} z_i \right]}{||\vec{R}'_z||} - 6t_z
\end{aligned} \tag{20}$$

As was mentioned earlier, we want to find a translation vector which averages the foot position coordinates. We want to have the resulting coordinate system O'_B lying in the averages of their coordinates with respect to the inclined regression plane. Therefore, the coordinates x_i and y_i are summed up to zero in the coordinate system O'_B. We can simply sibstitute

$$\begin{aligned}
\sum_{i=1}^{6} x'_i &= 0 \\
\sum_{i=1}^{6} y'_i &= 0
\end{aligned} \tag{21}$$

to the previous equation.

There comes the advantage of choosing a rotated translation vector $R\vec{t}$ instead of $\vec{t}$ in Equation (1) because we obtained a very simple expression of such a vector in the inverse transformation that can be implied from Equation (20). We conclude that

$$\begin{aligned}
t_x &= \frac{\sum_{i=1}^{6} x_i + a \sum_{i=1}^{6} z_i}{6 \cdot ||\vec{R}'_x||} \\
t_y &= \frac{-ab \sum_{i=1}^{6} x_i + (a^2+1) \sum_{i=1}^{6} y_i + b \sum_{i=1}^{6} z_i}{6 \cdot ||\vec{R}'_y||}.
\end{aligned} \tag{22}$$

Although the t_z coordinate of the translation vector $\vec{t}$ could be also computed similarly as the two other coordinates from Equation (20), we need to include the hexapod height h above the ground which has to be the same in both coordinate systems. This can be done a bit simpler following the description in Figure 5.4.

The regression plane $z = ax + by + c$, which is drawn here, shows us a right triangle (dashed) with it's two sides sharing the same angle between them as is between the z''-axis and the original z-axis. Because the z''-axis creates another right triangle with the unit vector of $\vec{b}_z$, these two triangles have the same angles and they are therefore similar. We can utilize this fact and obtain a simple equation of the ratio of the triangle's sides lengths as

$$\frac{1}{||\vec{b}''_z||} = \frac{1}{||\vec{R}'_z||} = \frac{t_z - h}{c}. \tag{23}$$

The coordinate t_z is thus simply expressed as

$$t_z = \frac{c}{||\vec{R}'_z||} - h. \tag{24}$$

After simplifying the notation using ($\sum \equiv \sum_{i=1}^{n}$) and substituting Equation (16), the resulting translation vector as it's coordinates are described in Figures 5.3 and 5.4 is therefore

$$\vec{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = \begin{bmatrix} \frac{\sum x_i + a\sum z_i}{6\sqrt{a^2+1}} \\ \frac{-ab\sum x_i + (a^2+1)\sum y_i + b\sum z_i}{6\sqrt{a^2b^2+a^4+2a^2+1+b^2}} \\ \frac{c}{\sqrt{a^2+b^2+1}} - h \end{bmatrix}. \tag{25}$$

5.5 Body Movement

The hexapod body motion is very dependent on the terrain it is crossing. If we consider an ideal flat environment[1]), the body motion can be deterministic although it's not constant.

There's no explicit forward body motion. The only body motion comes in the last step of the blind-tread gait (see diagram in Figure 5.1) when the body is leveled to the average of the x-coordinates of the foot positions. This average has been moved forward, because one (active) leg has moved forward. The body is therefore following the legs.

The leg stroke S is not as simply defined as in the ordinary gaits described in Section 3.5. It varies in every next step. Assuming an ideal environment, the leg stroke can be described by a schema in Figure 5.5.

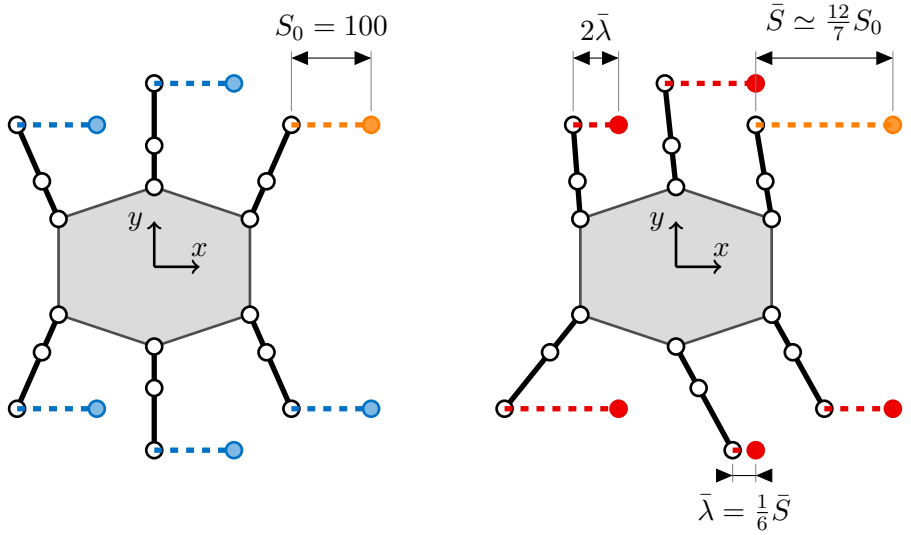


Figure 5.5. Dependencies between the leg stride λ, the initial leg stroke S_0 (left picture) and the final asymptotic stroke value $\bar{S}$ (right picture) which is reached after a few gait cycles in an ideal flat environment.

The left picture shows a hexapod in a default initial position. The blue marks represents the footsteps for each leg at the end of it's transfer phase. The footstep marks are sticked with the hexapod body which means that when a leg is chosen as an active leg, it moves up, forward and down to reach exactly the marked position in the

[1]) In an ideal environment, all leg positions are the desired ones and the ground detecting is 100% accurate which means that when a hexapod is running in place, the legs keeps the same positions after performing their motions. The leg trajectory is also a straight line going through the default leg position.

transfer phase. So, at the beginning of the support phase, the active leg is always in the same (marked) position with respect to the body (in an ideal environment). The initial distance between the default position and the next footstep position is the same for all legs and the leg stroke S which travels this distance is initially $S_0 = 100$ mm in the example in Figure 5.5. The distance S_0 can be set arbitrarily, but the possible leg collisions have to be handled carefully.

Since the next footstep positions are sticked with the body, the distance traveled by an active leg during it's transfer phase varies with increasing number of steps performed. This effect is described in the right picture of Figure 5.5. The red marks holds the same footsteps with respect to the body as in the left picture. But after leaving the initial position, the actual foot positions differ. In the right picture, a snapshot of a hexapod in a stable state of the gait cycle (see diagram in Figure 5.1) after five complete gait cycles performed with it's left front leg (orange) chosen as an active leg is shown here. The distance the left front leg has to travel has converged to the value of[1])

$$\bar{S} = \frac{12}{7} S_0, \tag{26}$$

where the leg stroke S_0 is the initial stroke value and hence the distance between the default leg positions and their corresponding next-step footmarks and $\bar{S}$ is the converged leg stroke value after performing several gait cycles.

The leg stride λ is completely dependent on the leg stroke S. Since the body motion is averaging the foot positions and the only position changed is the active-leg position, the leg stride is exactly one sixth of the leg stroke. Together with the leg stroke $\bar{S}$, it converges to

$$\bar{\lambda} = \frac{1}{6}\bar{S} = \frac{2}{7} S_0, \tag{27}$$

where $\bar{\lambda}$ is the converged stride value after performing several gait cycles.

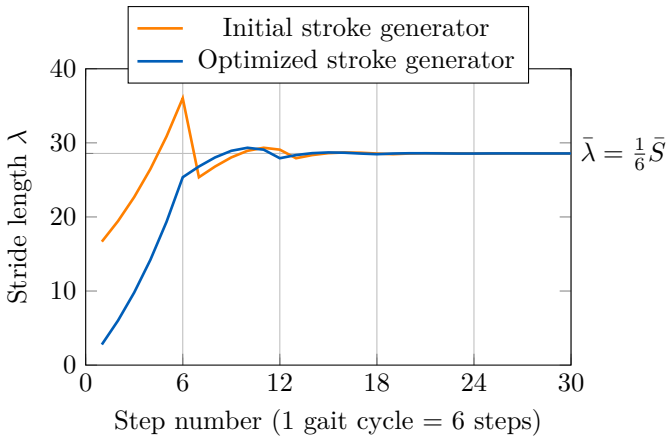


Figure 5.6. Dependency between the leg stride λ and the number of steps performed. The stride length value converges after a few gait cycles.

The graph of the converging leg stride λ to $\bar{\lambda}$ is shown in Figure 5.6. We can see that the initial value of λ in the first step is equal $\lambda_1 = 100/6 = 16.666$. As the body

[1]) This value is computed experimentally and without a proof. For an arbitrary number of legs n, this value converges obviously to $\bar{S} = S_0 \times 2n/(n+1)$. A proof can be given for such a gait with two legs only, but the complexity of a proof grows disproportionately with the number of legs.

moves forward, the legs are staying back and therefore their stroke increases to get to their next footstep. When all of the legs have moved at least once, the graph of λ discretely changes as the first leg moves the second time. This is because the leg stride λ is dependent on the last six moves which the current λ is averaging. We can say that

$$\lambda_n = \frac{1}{6} \sum_{i=n-6}^{n-1} \lambda_i, \tag{28}$$

for $n > 6$. Furthermore, we can add a λ_0 to cover all steps within a single equation

$$\lambda_i = \begin{cases} 0 & \text{for } i < 0 \\ S_0 & \text{for } i = 0 \\ \frac{1}{6} \sum_{i=n-6}^{n-1} \lambda_i & \text{for } i > 0 \end{cases}. \tag{29}$$

Since this discontinuity leads to a behavior we don't want to achieve, a simple stroke generator was applied to improve the graph of λ. The first six steps are suppressed in a following rate

$$S_i = \begin{cases} \frac{i}{6} S_0 & \text{for } i < 6 \\ S_0 & \text{for } i \geq 6 \end{cases}. \tag{30}$$

This improvement gives us a better behavior of the blind-tread gait in the first two gait cycles. We can see in Figure 5.6 that the stride length λ converges very fast and there can hardly be observed any discontinuities after completing three gait cycles.

As can be seen from the right picture of Figure 5.5, the distance between each leg and it's next footstep in each stable state of the blind-tread gait is equal to a multiple of $\bar{\lambda}$ corresponding to a given leg order.

In real situations, the leg stroke and stride are not deterministic. They vary depending on the footstep position that a leg has found in it's transfer phase. The leg positions can differ slightly when crossing a rough terrain, so the initial stroke value S_0 cannot be set very high.

The robot can also move and rotate his body in all six dimensions in addition to leveling the body. But this additional transformation is independent of the transformation described above, so it isn't included in the gait diagram in Figure 5.1 although it uses the leg positions transformation for body movement too.

The body position can be moved forward with respect to it's base coordinate system O_B and not affecting all of the actions described in this chapter at all. But it can give an advantage when moving a sloped terrain to shift the COG to a position where the stability of the robot is higher. Shifting the body position also negatively influences the leg working space, so the stroke value S_0 is limited when improving the COG position.

Chapter 6
Experimental Results

The blind-tread gait performance was tested on various terrains under various walking speeds. The experiments were performed on the same robot as was described in previous chapters without any additional sensors to measure the gait performance. The results are visualized here using the robot's internal data only. All internal data were written during the experiments while not affecting the robot's behavior (e.g. the speed). Only in the steady states between the leg steps, all joint angles were read in a pack which increased the total experiment time by about 5%.

Although the robot's parameters were set in a way to increase the gait speed, there were some limitations that couldn't have been exceeded. The limits come mainly from the actuators which can't operate in a high speed with a compliance set to the highest value and keep the motion smoothness at the desirable level both at the same time.

The resulting speed of the gait (not the speed of the robot) is about two and a half minutes per ten complete gait cycles (no matter what distance the robot has actually traveled).

6.1 Steady Performance

Firstly, we have to verify whether the developed gait meets the expectations of a gait that can handle operations in a rough environment. The simplest rough terrain is a flat ground. A result from a steady experiment on a flat ground where the legs are running in place, i.e., there's no forward movement, is shown in Figure 6.1.

Three different controllers were used to control the body position of the hexapod during ten complete gait cycles performed. The compared controllers share the same ground detection mechanism (explained in Section 5.3) but they differ in the way they handle the positions transformation. The methods are

- No transformation – each leg keeps it's position where it detected the ground.
- Average Z – all legs are shifted such that the average of their z-coordinates is equal to the default hexapod height h above the ground.
- Transformation – all legs are transformed following the method described in Section 5.4.

We can see that when no transformation is used, the hexapod is falling to it's left side. When a z-averaging method is used, the height h above the ground is kept constant, but the falling to the left side is not avoided. Finally, a developed transformation method keeps all legs at the same elevation level in a long time horizon. The effort which keeps the legs leveled can be seen from a frequent adjustments to the leg positions (orange), occasional adjustments (blue) and no adjustments between the leg's active cycles at all (red).

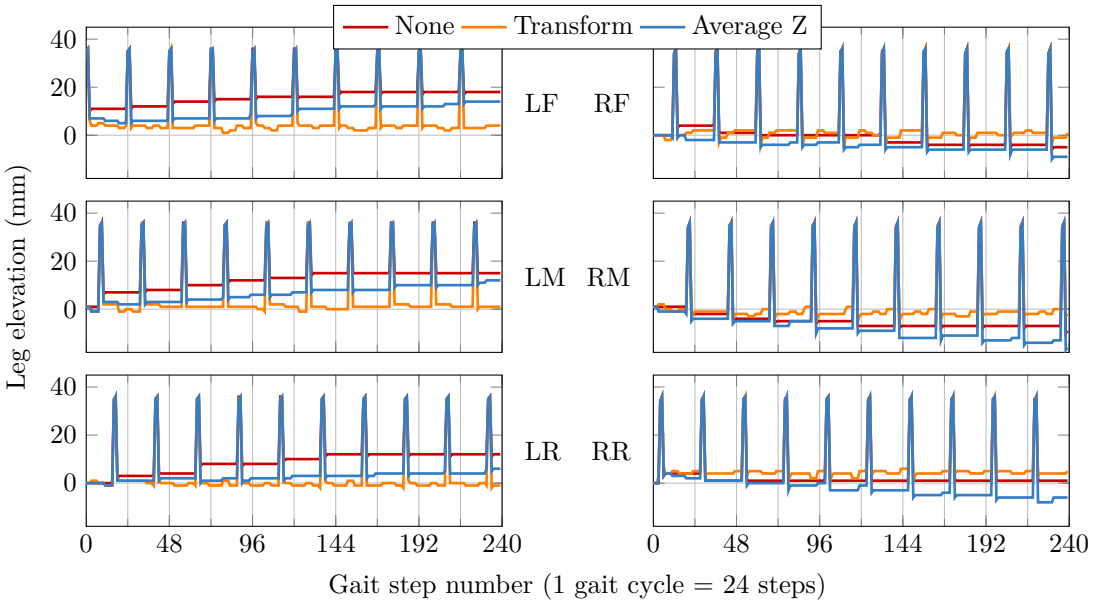


Figure 6.1. Flat ground comparison between different controllers used when a leg reaches the ground. Data is gathered from all legs during 10 complete gait cycles on a flat ground when horizontal movement is suppressed (legs are running in place). Transformation controller is the only one which keeps the legs balanced and thus, the body is not tilted.

6.2 Key Performance Indicator

The performance measure which reflects the gait performance is very hard to find since we have a limited data available. Having an accelerometer or force sensors would be a good solution, but the hexapod should have utilized these sensors itself too. Considering what data are actually available, we end with a set of actual and desired foot positions, a set of actual and desired joint angle positions and a set of computed parameters (a, b and c of the regression plane and t_x, t_y and t_z of the translation vector).

Imagining the dependencies between each of these parameters and positions and the quality of the gait, we don't have a clear winner. Let us think about what we tend to achieve. One of the main goals of the developed gait is that it can smoothly traverse a rough terrain. The difference between this and the default gaits presented in Section 3.5 is that the default gaits have always the foot positions of all support legs lying in the same plane. Although they can traverse a slightly rough terrain, there will always be a leg in a support phase which is actually moving in the air, i.e., not supporting the robot at all. This is the area where the blind-tread gait can compete very well.

A good performance indicator could therefore be a visualization of the support function of all legs. We will be expecting a graph where all of the legs will be more or less supporting the robot. The measure of support level can be approximated by the foot position displacement in the z-axis as is discussed in Section 4.5. This value approximates the resulting ground reaction force quite authentically (the best among all of the data available). We can also use the femur joint angle displacement (as is used in the detection of the ground in Section 5.3) but the force the femur angle displacement corresponds to has a different direction compared to the z-axis displacement. It has a better resolution, but the lower resolution of z-axis will be enough.

Thus, the z-axis displacement of all legs will be used to measure the performance of the blind-tread gait in all of the experiments below.

6.3 Scenarios

All scenarios of the experiments are shown in Figure 6.2.

Firstly, the gait performance was tested on a flat ground (a) where the results are not affected by the terrain difficulties. Every single inaccuracy can be spotted here. The hexapod has to prove here that the gait is designed general enough to walk smoothly on an even terrain.

Secondly, a slightly sloped terrain (b) with angle of 10° was set to prove the hexapod ability to move smoothly over a transition to an inclined plane. The inclined plane is made from wood on which the hexapod hasn't a good adhesion. Due to the frequent slippages, this terrain isn't so easy despite it's low angle.

Thirdly, a greater angle of 20° of an inclined plane (c) was set in order to check how far the hexapod can go on a sloped terrain. Since it is the same piece of wood as in (b), this terrain is even more slippery due to it's angle. The position of the COG plays an important role here.

Fourthly, a stairs (d) with the height of each step about 5 cm and the same slope as in (c) was set to examine the hexapod on even more challenging terrain.

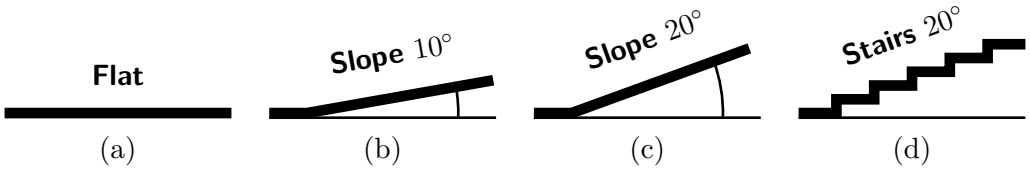


Figure 6.2. Testing scenarios for all experiments.

6.4 Testing Stride Length

There are many parameters (as usually) that can be optimized all the time. Some of them are set and never changed because they have a negligible influence on the overal performance, some of them are set and never changed because they are crucial and changing their value would cause the loss of stability for example, some of them has influence on too many factors so it is very hard to find their right value mostly because they have different right values for different environments.

Among all of the parameters that can be changed on hexapod, we chose to make experiments with variable stride length (by setting the initial leg stroke S_0 – described in Section 5.5) because such an experiment indicates the true variability in a hexapod motion.

Along with the stride length, the body position forward offset has also been changing during the experiments mostly in order to compensate the COG position which was inclined by a sloped terrain. Such a change was never made on the fly but before the experiment started since we knew what terrain the hexapod is going to deal with.

A set of figures for each scenario is shown in the following sections. A graph of the leg z-axis displacement is shown in each figure for all legs and the area between the graph and the zero value is filled to highlight the actual value of leg displacement in time (negative displacement which indicates that the leg is not supporting is highlighted in red). Because the leg positions are most of the time in the supporting phase constant and they're changed only in the body leveling step, the values measured were gathered after each leveling step only. Thus, the gait cycle contains six steps in which all legs should lay on the ground.

Each scenario contains four experiments each with growing initial stroke (S_0 = 30, 40, 50 and 60 mm).

6.4.1 Flat Ground

The experiments performed on a flat ground are shown in Figures 6.3, 6.4, 6.5 and 6.6.

Firstly, we can see that the ground reaction force is not uniformly distributed between the legs. Although all of the legs share the same controller, for example the right middle leg and partially the right front leg are always supporting less than the left middle and front leg. If we look in more detail at the first few steps, we can see that the right middle leg always starts with a negative support level. The only reason can be that the hexapod legs are not completely symmetric. The rear legs carry the greatest portion of support and as can be seen from these figures, they alternate in carrying significant part of the weight of the robot.

Despite the irregularities, we can conclude that all of the legs are more or less supporting the robot all the time at all tested walking stride lengths.

6.4.2 Inclined Slope (10°)

The results of a transition from a flat ground to a 10° inclined slope are shown in Figures 6.7, 6.8, 6.9 and 6.10.

In this scenario, the hexapod made the first two gait cycles on a flat ground before reached an inclined plane. The time the hexapod spent traversing the transition between these two terrains varies with the walking stride length. In the first experiment with the lowest value of stride, the hexapod reached the inclined slope completely by all legs just at the end of the last gait cycle. The time needed to traverse the transition is roughly linearly dependent on the stride length, so in the last experiment with the highest value of stride, the last several gait cycles were performed completely on the inclined plane.

The performance is similar to the previous scenario although the total measured support level is lower, but this could be due to the sloped terrain on which the coxa joints shares a portion of support too (but they're not measured). Nevertheless, we can see that the observations from previous scenario about the right front and middle legs came true. Especially the right front leg is hardly supporting the robot (comparing to the left front leg). With higher walking strides and with greater body tilt, the right front leg is supporting less which can be interpreted as the result of the slope of the terrain which shifts the COG above the rear legs (which can be shown on the graphs of rear legs support level).

6.4.3 Inclined Slope (20°)

The results of a transition from a flat ground to a 20° inclined slope are shown in Figures 6.11, 6.12, 6.13 and 6.14.

When compared to the previous scenario of half as slope terrain, we can't directly see much differences except that the total measured support level is a bit lower which is, as was said before, due to the direction of the ground reaction force partially covered by the coxa joints.

The fact that can't be seen from these figures is that the terrain is slippery for the hexapod. The slippage occurrence is frequent mainly in the case of the front legs. The slippage occurs when a leg is hitting the ground and increases it's support level in order to achieve the ground detection threshold value (described in Section 5.3). In this moment, there is no weight left that could be supported by the other adjacent legs

which are therefore elevated enough to lose their friction with the plane and the leg is slipping down the plane due to the compliance in the coxa joints (although the coxa joints are set as stiff as possible – see Section 4.1).

As in the previous scenario, we can conclude that the level of support of individual legs follows the expected values as the main support is made by the rear legs while the front legs try not to lose their (minimal) support level.

6.4.4 Stairs

The stairs scenario is the complex one. It has an advantage that there is no slippage down the slope as can be seen in the case of an inclined plane. But there is a great disadvantage for a blind robot that he doesn't know where a stair actually begins. The worst case is when a leg finds it's foothold on the edge of a stair and, few steps later when the support level of this leg increases, the foot falls to the previous step and the stability of whole hexapod is severely disrupted. This happened many times during the experiments shown in Figures 6.15, 6.16, 6.17 and 6.18 but every time it didn't take a lot of time for the hexapod to successfully recover to the full support operation. The hexapod managed to climb the stairs although he had no idea about the terrain he was crossing.

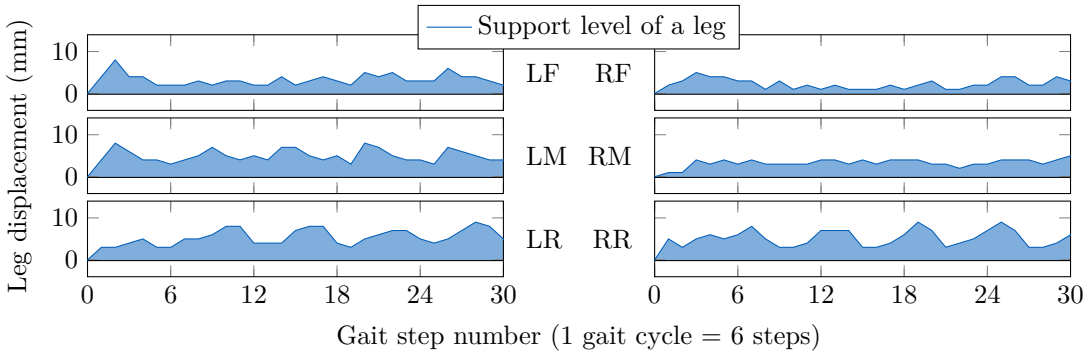


Figure 6.3. Support level of all legs while walking on a flat ground with $S_0 = 30$ mm.

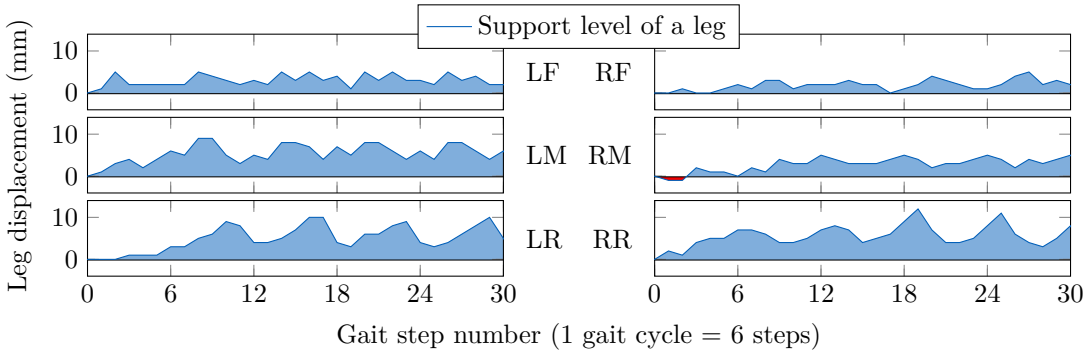


Figure 6.4. Support level of all legs while walking on a flat ground with $S_0 = 40$ mm.

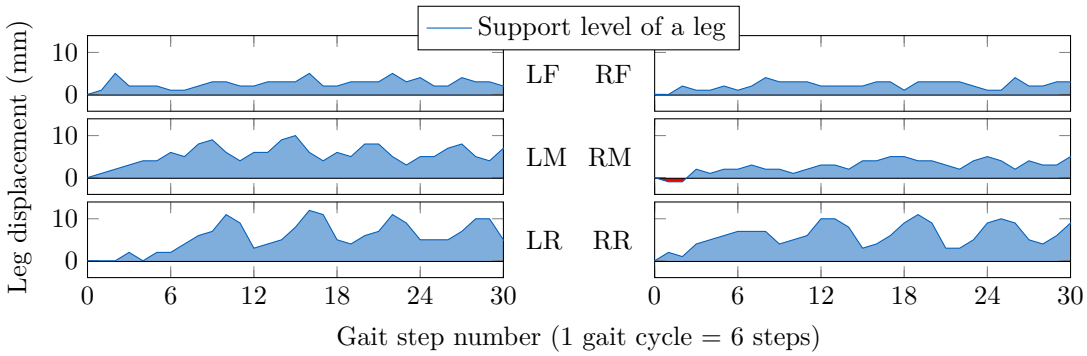


Figure 6.5. Support level of all legs while walking on a flat ground with $S_0 = 50$ mm.

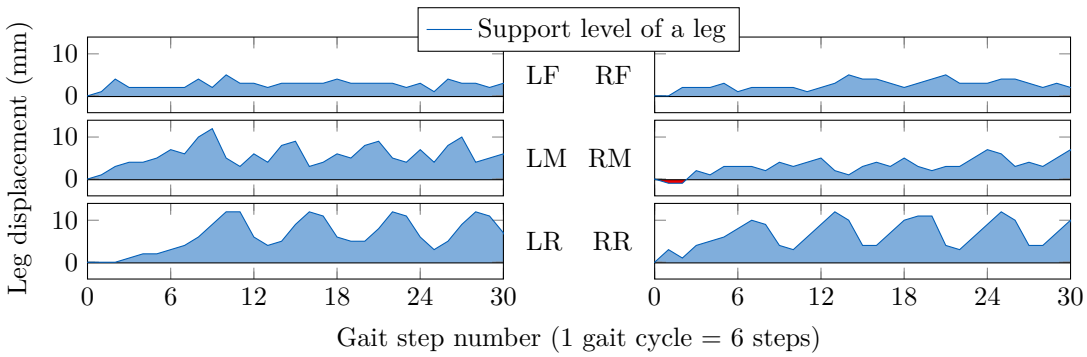


Figure 6.6. Support level of all legs while walking on a flat ground with $S_0 = 60$ mm.

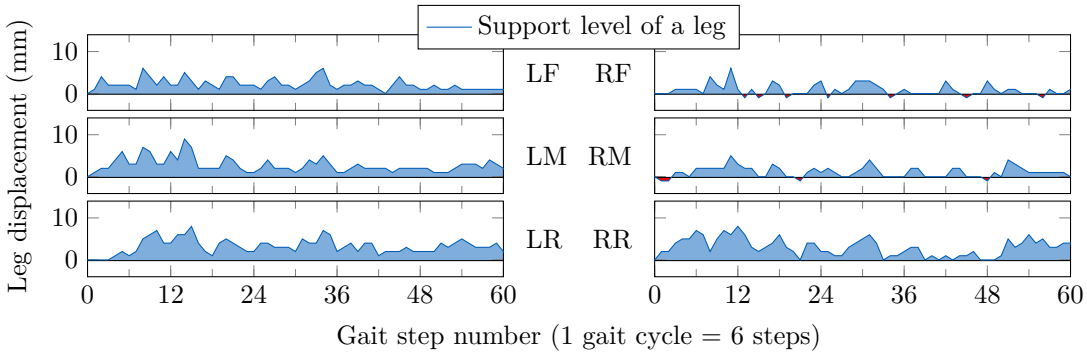


Figure 6.7. Support level of all legs while walking on a 10° slope with $S_0 = 30$ mm.

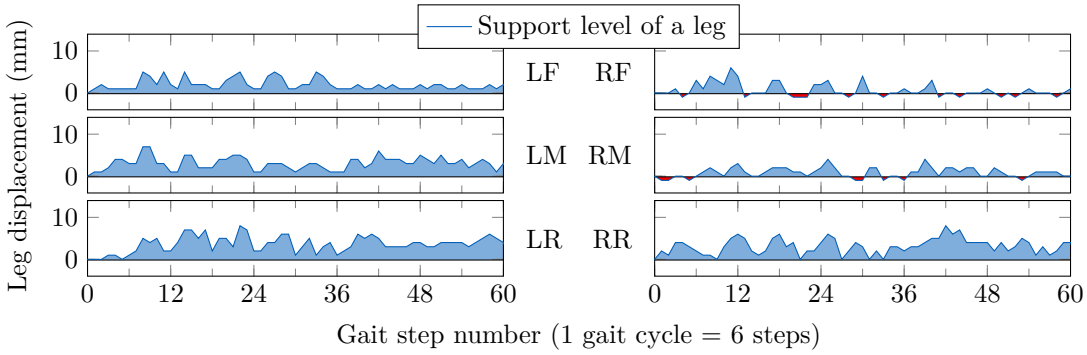


Figure 6.8. Support level of all legs while walking on a 10° slope with $S_0 = 40$ mm.

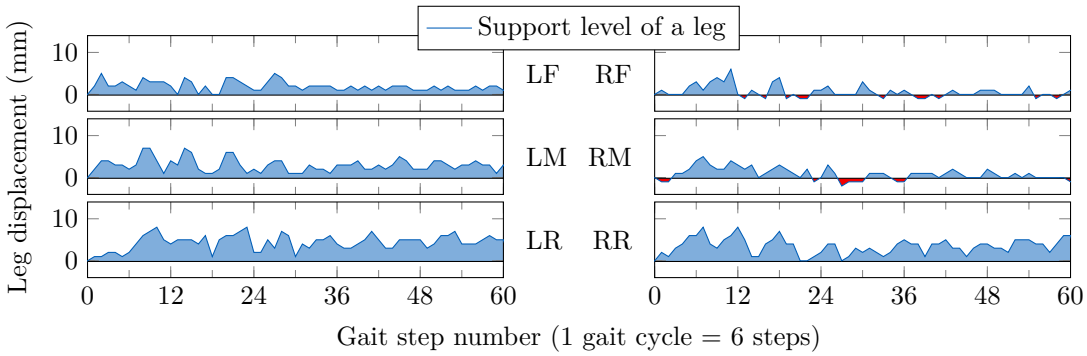


Figure 6.9. Support level of all legs while walking on a 10° slope with $S_0 = 50$ mm.

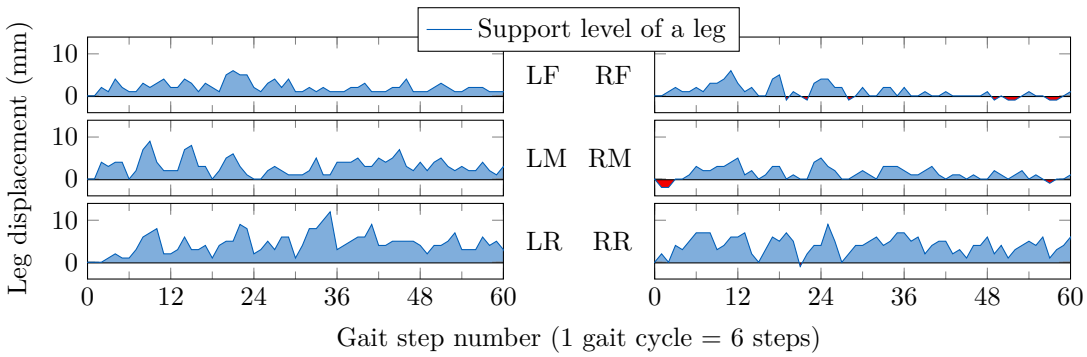


Figure 6.10. Support level of all legs while walking on a 10° slope with $S_0 = 60$ mm.

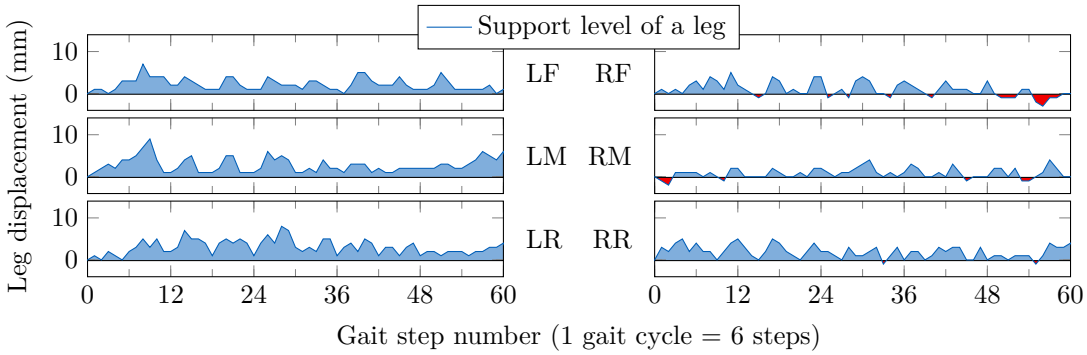


Figure 6.11. Support level of all legs while walking on a 20° slope with $S_0 = 30$ mm.

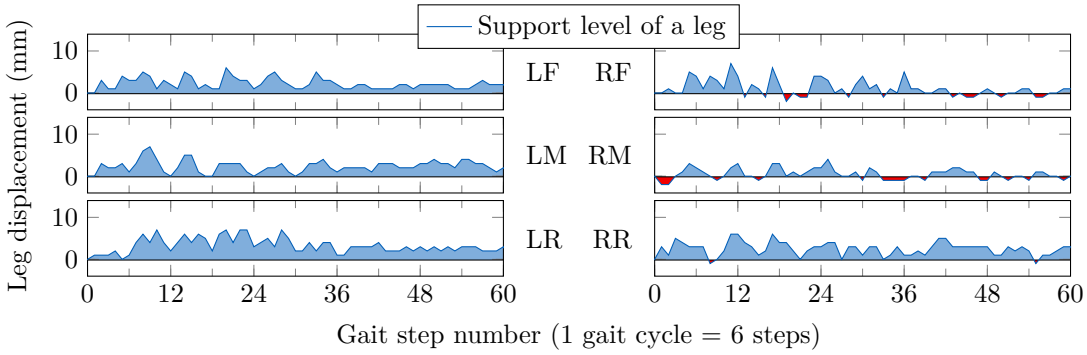


Figure 6.12. Support level of all legs while walking on a 20° slope with $S_0 = 40$ mm.

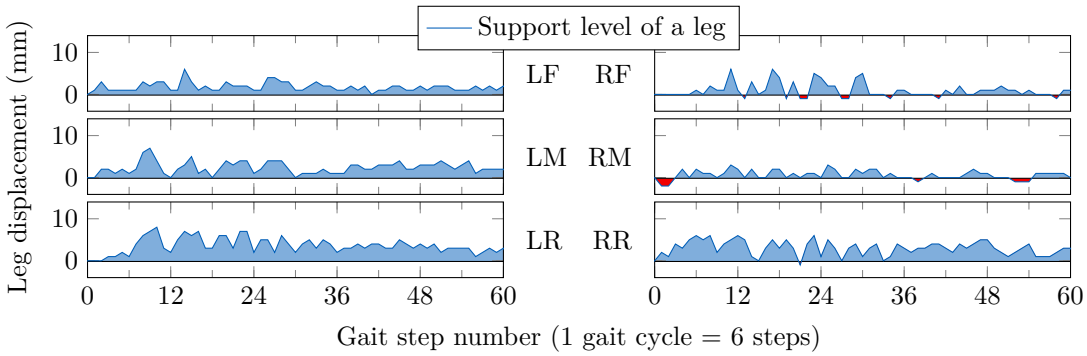


Figure 6.13. Support level of all legs while walking on a 20° slope with $S_0 = 50$ mm.

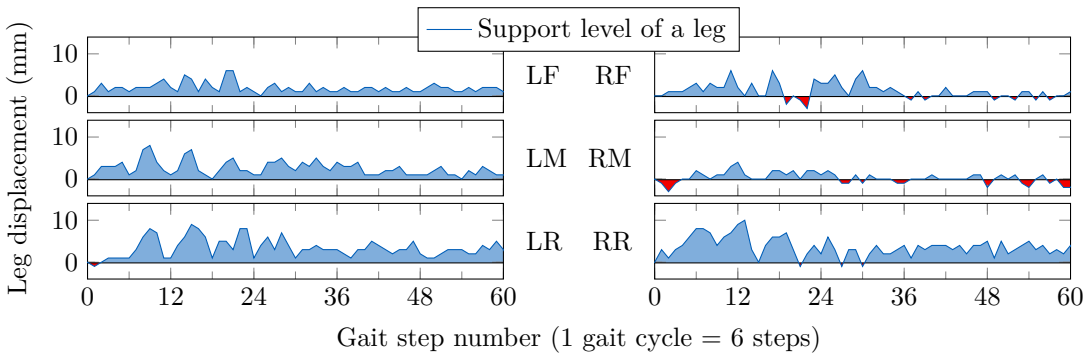


Figure 6.14. Support level of all legs while walking on a 20° slope with $S_0 = 60$ mm.

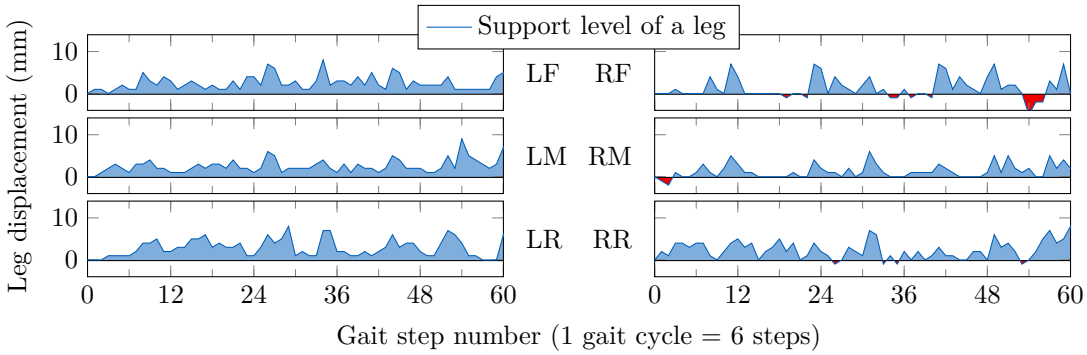


Figure 6.15. Support level of all legs while climbing stairs with $S_0 = 30$ mm.

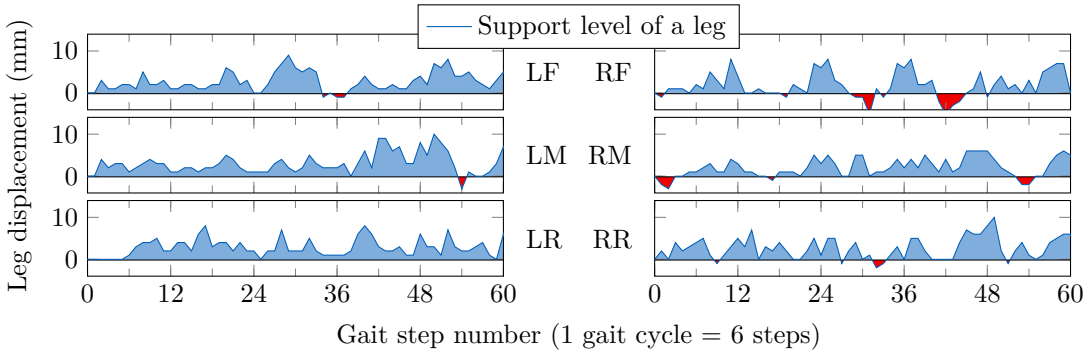


Figure 6.16. Support level of all legs while climbing stairs with $S_0 = 40$ mm.

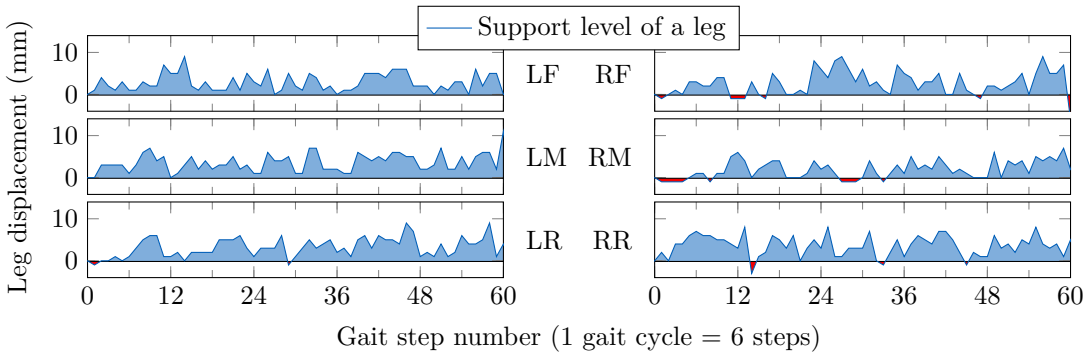


Figure 6.17. Support level of all legs while climbing stairs with $S_0 = 50$ mm.

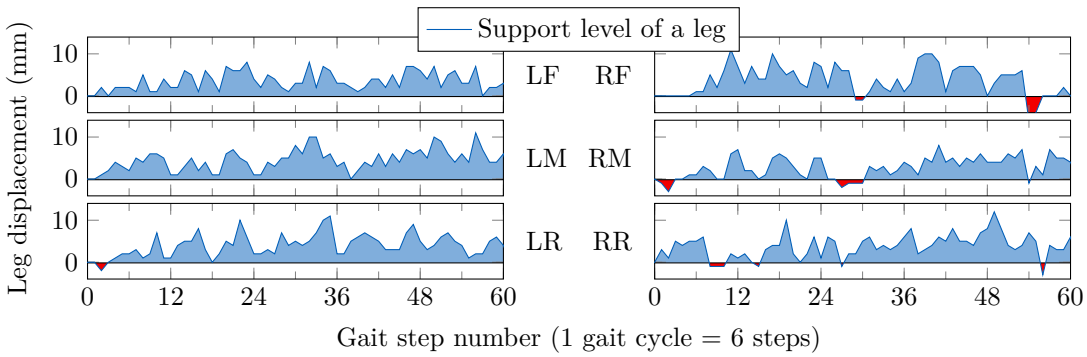


Figure 6.18. Support level of all legs while climbing stairs with $S_0 = 60$ mm.

Chapter 7
Discussion

A hexapod has a great potential in mobile robotics. It's six legs allows him to utilize a variety of different gaits for different purposes like using a tripod gait for a fast motion over a flat terrain, switching to the blind-tread gait when an unstructured transition needs to be traversed, then apply a stairs-climbing gait, etc.

Although a hexapod is very popular, there are octopods or even a centipede inspired robots with many legs already. More legs brings a great operational variability to a legged walking robot. Besides all of the actions he can perform due to it's numerous legs, he might even be able to self-repair his current gait when a leg breaks down in order to regain it's walking stability.

7.1 Tactile Sensing

Another extensive area is the tactile perception. Assuming that the robot has a similar ground detection mechanism as the one described in Section 5.3 implemented, gaining an information about the actual leg positions opens up a new dimension for him if the robot doesn't have any scanning sensors like range-finder or camera. In the developed gait in Chapter 5, the acquired foot positions on an uneven terrain are used only once and then they are forgotten. If they were kept in the memory with a precise link to the current robot's position (which implies an accurate odometry), the robot could have it's own footstep map.

Besides the fact that the robot can trace back it's own trajectory, it's motion can be used to create such footsteps primarily in order to map the terrain around the robot. Since the robot has numerous legs, the number of footsteps will not be small. Assuming that the robot has a particular goal, he can chose which area to map more densely and which loosely according to the terrain smoothness. In order to increase the precision of the mapping, he can stay at one place with only one leg in the air patiently scanning the terrain by touching the surface around the robot.

The map created from the footsteps can be corrected using additional information from other sensors, or it can be used to correct the robot's mapping accuracy by visiting the mapped positions multiple times.

The tactile sensing can also be used in a careful manipulation. When such a gait is applied that the robot has one or more legs free while still able to walk, he can become an ant-like robot using it's legs to find, pick up and drag items.

7.2 Unstructured Environment

Operating in an unstructured environment requires a general gait that can handle all unperceived obstacles and terrain difficulties. We can divide the term of unstructured environment into three main areas that are graded with increasing difficulty of the environment.

- A static environment with arbitrary shape of the surface.
- The same as above but with considering adhesion (muddy, wet surfaces).
- A dynamic environment with outer forces acting on the robot.

The blind-tread gait presented in this work is able to handle the first type (level) of unstructured environment. It has a little problems with a slippery terrain. After a slippage occurs, most of the legs stay in their current configuration and they are waiting for their time to move. An ideal case would be measuring the foot displacement online for all legs at a time (as in the work of Palankar et al. [18–19]). But this approach would require a custom hardware with each leg handled by an individual controller instead of a mass produced robot. Online reading can lead to a dynamic gait that has a potential to be able to handle the third type of environment if the controllers of the legs are continuously setting appropriate positions in order to keep the leg support at the same (desired) level.

7.3 Future Work

The current state of the blind-tread gait presented in Chapter 5 has been successfully experimentally tested on walking on a flat and sloped terrain (with slope up to 20°), traversing transitions between such terrains and even climbing the stairs.

There are many possible upgrades that can improve it's performance. Among the first could be increasing precision and robustness of the ground detection mechanism by using more than a single servo angle displacement.

Another area that needs to be explored well is the leg working space. The working space has been described in this work but further work is needed to cover the sensitive area of possible leg collisions and increase the reach of the legs which is now very limited.

Experiments with various loads on the robot could also give us another important information about the robot's capabilities. In this work a fully unloaded hexapod was tested only.

In the case of adding external sensors, an accelerometer could help the robot with better fitting the body position forward offset in order to compensate the shifted COG when walking up or down the slope.

Another important area is a self-calibration. An accelerometer can help here a lot by detecting and compensating the body leveling errors on a flat ground. The legs are not balanced in the factory settings. When a hexapod is staying on a flat ground, each leg detects a different z-axis displacement which could be corrected by getting the position information from another sensor.

Chapter 8
Conclusion

Starting with an ordinary remote controlled hexapod walking robot without any external sensors, a newly developed walking gait called *Blind-Tread Gait* was applied that adds to the hexapod the ability to walk successfully over a rough terrain environment.

The hexapod is technically blind and can only utilize it's actuator positions. After studying the core of the hexapod and the limits of it's actuators, a motion sub-primitive was developed that allows the hexapod to detect the ground during a leg motion primitive within a gait cycle. Such detected positions are used as a footsteps for hexapod legs in successive gait steps.

The body position and orientation is computed using the body-leveling transformation described in Section 5.4 which ensures that the body moves to an equilibrium position with respect to the leg positions after every leg motion. The resulting body motion is described and analyzed in detail as well as the leg stroke.

The developed gait was examined in several experiments. It's performance was measured on a graph of support level of each leg while the hexapod was walking on a flat terrain, sloped terrain and climbing the stairs. All experimental scenarios were repeated with different walking stride lengths. The results show us that the newly developed gait's performance does not significantly decrease with a more challenging terrain or a greater walking stride length. The gait is so general that the hexapod can even climb the stairs quite smoothly. Although he occasionally slips from a stair's edge (because he is still blind), the recovery time is very fast despite the fact that there's no explicit recovery mechanism.

References

[1] Marc Reibert, Kevin Blankespoor, Gabriel Nelson, Rob Playter, and the BigDog Team. Bigdog, the rough-terrain quaduped robot. In *Proceedings of the 17th International Federation of Automation Control*, April 2008. `http://www.bostondynamics.com/img/BigDog_IFAC_Apr-8-2008.pdf`.

[2] Mrinal Kalakrishnan, Jonas Buchli, Peter Pastor, Michael Mistry, and Stefan Schaal. Learning, planning, and control for quadruped locomotion over challenging terrain. *The International Journal of Robotics Research*, 30(2):236–258, February 2011.

[3] Adam Schmidt and Krzysztof Walas. The classification of the terrain by a hexapod robot. In Robert Burduk, Konrad Jackowski, Marek Kurzynski, Michał Wozniak, and Andrzej Zolnierek, editors, *Proceedings of the 2013 8th International Conference on Computer Recognition Systems CORES*, volume 226 of *Advances in Intelligent Systems and Computing*, pages 825–833. Springer International Publishing, May 2013.

[4] Krzysztof Walas. Tactile sensing for ground classification. *Journal of Automation, Mobile Robotics & Intelligent Systems*, 7(2):18–23, June 2013.

[5] Krzysztof Walas. Terrain classification using vision, depth and tactile perception. In *Proceedings of the 2013 RGB-D: Advanced Reasoning with Depth Cameras in conjunction with RSS*, June 2013. `http://mobilerobotics.cs.washington.edu/rgbd-workshop-2013/`.

[6] Krzysztof Walas. Improving accuracy of local maps with active haptic sensing. In Krzysztof Kozlowski, editor, *Robot Motion and Control 2011*, volume 422 of *Lecture Notes in Control and Information Sciences*, pages 137–146. Springer London, 2012.

[7] Krzysztof Walas. Foot design for a hexapod walking robot. *Pomiary, Automatyka, Robotyka*, 17(2):283–287, February 2013.

[8] Krzysztof Walas and Dominik Belter. Supporting locomotive functions of a six-legged walking robot. *International Journal of Applied Mathematics and Computer Science*, 21(2):363–377, June 2011.

[9] William A. Lewinger. Insect-inspired, actively compliant robotic hexapod. Master's thesis, Case Western Reserve University, May 2005.

[10] Jie Zhao, He Zhang, Yubin Liu, Jihong Yan, Xizhe Zang, and Ziwei Zhou. Development of the hexapod robot HITCR-II for walking on unstructured terrain. In *Proceedings of the 2012 International Conference on Mechatronics and Automation (ICMA)*, pages 64–69, August 2012.

[11] Katie Lynn Hoffman. *Design and locomotion studies of a miniature centipede-inspired robot*. PhD thesis, Harvard University, May 2013.

[12] Sangbae Kim. *Bio-inspired robot design with compliant underactuated systems*. PhD thesis, Stanford University, March 2008.

[13] Junke Li, Yujun Wang, and Ting Wan. Design of a hexapod robot. In *Proceedings of the 2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet)*, pages 1768–1771, April 2012.

[14] Donald Campbell. Bounding and stair descent in the hexapod RHex. Master's thesis, McGill University (Canada), February 2004.

[15] Bir Bikram Dey, Sandeep Manjanna, and Gregory Dudek. Ninja legs: Amphibious one degree of freedom robotic legs. In *Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5622–5628, November 2013.

[16] Travis Thomson. Kinematic control and posture optimization of a redundantly actuated quadruped robot. Master's thesis, McGill University (Canada), December 2011.

[17] Graeme Best, Peyman Moghadam, Navinda Kottege, and Lindsay Kleeman. Terrain classification using a hexapod robot. In *Proceedings of the Australasian Conference on Robotics and Automation*, December 2013. `http://www.araa.asn.au/acra/acra2013/papers/pap155s1-file1.pdf`.

[18] Mayur Palankar. *A distributed local-leg feedback algorithm for robust walking on uneven terrain*. PhD thesis, University of South Florida, May 2013.

[19] Luther Palmer and Mayur Palankar. Blind hexapod walking over uneven terrain using only local feedback. In *Proceedings of the 2011 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1603–1608, December 2011.

[20] Benjamin B. Choi. *Wave gaits, follow-the-leader gaits and gait transitions of legged robots*. PhD thesis, University of Illinois at Chicago, 1990.

[21] Jung-Min Yang. Fault-tolerant gait planning for a hexapod robot walking over rough terrain. *Journal of Intelligent and Robotic Systems*, 54(4):613–627, April 2009.

[22] Martin Görner and Annett Stelzer. A leg proprioception based 6 DOF odometry for statically stable walking robots. *Autonomous Robots*, 34(4):311–326, May 2013.

[23] Trossen Robotics. *PhantomX Hexapod Mark II Assembly Guide*. `http://trossenrobotics.com/productdocs/assemblyguides/PhantomX-Hexapod-Mark2.html`.

[24] ROBOTIS. *ROBOTIS AX-18 Actuator e-Manual v1.15*. `http://support.robotis.com/en/product/dynamixel/ax_series/ax-18f.htm`.

[25] Advait Jain. *Mobile manipulation in unstructured environments with haptic sensing and compliant joints*. PhD thesis, Georgia Institute of Technology, December 2012.

[26] Richard S. Hartenberg and Jacques Denavit. A kinematic notation for lower pair mechanisms based on matrices. *Journal of Applied Mechanics*, 77:215–221, June 1955.

[27] Michael Mistry, Jonas Buchli, and Stefan Schaal. Inverse dynamics control of floating base systems using orthogonal decomposition. In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3406–3412, May 2010.

[28] Michael Mistry, Jun Nakanishi, Gordon Cheng, and Stefan Schaal. Inverse kinematics with floating base and constraints for full body humanoid robot control. In

Humanoids 2008. 8th IEEE-RAS International Conference on Humanoid Robots, pages 22–27, December 2008.

[29] Michael Mistry, Stefan Schaal, and Katsu Yamane. Inertial parameter estimation of floating base humanoid systems using partial force sensing. In *Humanoids 2009. 9th IEEE-RAS International Conference on Humanoid Robots*, pages 492–497, December 2009.

[30] Jun Nakanishi, Michael Mistry, and Stefan Schaal. Inverse dynamics control with floating base and constraints. In *Proceedings of the 2007 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1942–1947, April 2007.

[31] David Eberly. *Least Squares Fitting of Data*, February 2008. `http://www.geometrictools.com/Documentation/LeastSquaresFitting.pdf`.

[32] Saunders MacLane and Garret Birkhoff. *Algebra*. Chelsea Publishing Company, 3rd edition, 1988.

[33] Tomáš Pajdla. *Elements of Geometry for Robotics*, November 2013. `http://cmp.felk.cvut.cz/cmp/courses/PRO/2013/Lecture/PRO-2013-Lecture.pdf`.

[34] Vladimír Smutný. *Geometry and algebra used in kinematics*, October 2013. `http://cmp.felk.cvut.cz/cmp/courses/ROB/roblec/geometry-noteeng.pdf`.

Appendix A
Specification

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Cybernetics

DIPLOMA THESIS ASSIGNMENT

Student: Bc. Jakub M r v a

Study programme: Cybernetics and Robotics

Specialisation: Robotics

Title of Diploma Thesis: Design of Motion Primitives for a Hexapod Walking Robot Operating in a Rough Environment

Guidelines:

1. Study the actuators used at the hexapod walking robotic platform.
2. Study available motion primitives.
3. Propose a decomposition of these primitives to sub-primitives.
4. Generalize motion primitives using designed sub-primitives that utilize feedback from actuators for motion control in a rough environment.
5. Develop a set of generalized motion primitives for selected crawling methods that will allow to move the robot from a plane to an inclined plane.
6. Discuss how the feedback from actuators can be used as tactile sensor.
7. Discuss possibilities how to extend the developed approach for walk in unstructured environment.

Bibliography/Sources:

[1] Steven M. LaValle: Planning Algorithms. Cambridge University Press, May 2006.
[2] Belter, D.; Łabęcki, P.; Skrzypczyński, P.: Map-based adaptive foothold planning for unstructured terrain walking. In Robotics and Automation (ICRA), 2010 IEEE International Conference on, 5256-5261.
[3] Belter, D.; Kasiński, A.; Skrzypczyński, P.: Evolving feasible gaits for a hexapod robot by reducing the space of possible solutions. In Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on , pp.2673,2678, 22-26 Sept. 2008
[4] Shibendu Shekhar Roy, Dilip Kumar Pratihar: Modeling of six legged robots: Analytical and soft computing-based methods. Lambert Academic Publishing, 2012.

Diploma Thesis Supervisor: Ing. Petr Vaněk

Valid until: the end of the summer semester of academic year 2014/2015

L.S.

doc. Dr. Ing. Jan Kybic
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, December 6, 2013

A.1 Czech Version

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra kybernetiky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: Bc. Jakub M r v a

Studijní program: Kybernetika a robotika (magisterský)

Obor: Robotika

Název tématu: Návrh pohybových primitiv pro šestinohý kráčející robot pohybující se v nerovném prostředí

Pokyny pro vypracování:

1. Seznamte se s akčními členy šestinohé kračející robotické platformy.
2. Seznamte se s dostupnými základními primitivy pohybu platformy.
3. Navrhněte dekompozici primitiv na pohybová subprimitiva.
4. Zobecněte primitiva pohybu s využitím subprimitiv tak, aby bylo možné využít zpětnou vazbu z akčních členů pro chůzi v nerovném prostředí.
5. Pro zvolené vzory pohybu vytvořte množinu primitiv, které umožní robotu plynulý pohyb z roviny na jinou, nakloněnou, rovinu.
6. Diskutujte jak může být využita zpětnovazební informace aktuátoru v úloze taktilní snímání pracovního prostředí robotu.
7. Diskutujte možnosti rozšíření navrženého přístupu pohybových primitiv se zpětnou vazbou pro pohyb robotu v nestrukturovaném prostředí.

Seznam odborné literatury:

[1] Steven M. LaValle: Planning Algorithms. Cambridge University Press, May 2006.
[2] Belter, D.; Łabęcki, P.; Skrzypczyński, P.: Map-based adaptive foothold planning for unstructured terrain walking. In Robotics and Automation (ICRA), 2010 IEEE International Conference on, 5256-5261.
[3] Belter, D.; Kasiński, A.; Skrzypczyński, P.: Evolving feasible gaits for a hexapod robot by reducing the space of possible solutions. In Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on , pp.2673,2678, 22-26 Sept. 2008
[4] Shibendu Shekhar Roy, Dilip Kumar Pratihar: Modeling of six legged robots: Analytical and soft computing-based methods. Lambert Academic Publishing, 2012.

Vedoucí diplomové práce: Ing. Petr Vaněk

Platnost zadání: do konce letního semestru 2014/2015

L.S.

doc. Dr. Ing. Jan Kybic
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 6. 12. 2013

Appendix B
Used Terms and Software

B.1 Abbreviations

DOF	Degree of Freedom.
COG	Center of Gravity.

B.2 Symbols

θ_C	The coxa joint angle.
θ_F	The femur joint angle.
θ_T	The tibia joint angle.
h	The hexapod height above the ground.
λ	Leg stride.
S	Leg stroke.
β	Duty factor.
O_i	Coordinate system.
$\theta_i, d_i, a_i, \alpha_i$	Denavit-Hartenberg parameters.
I	Identity matrix.
$A^{4\times4}$	Tranformation matrix in homogeneous coordinates.
$R^{3\times3}$	Rotation matrix.
a, b, c	Parameters of the regression plane $z = ax + by + c$.
t_x, t_y, t_z	The coordinates of the body-leveling translation vector $\vec{t} = [t_x, t_y, t_z]^T$.
$\|\|\vec{x}\|\|$	Euclidean norm of $\vec{x}$ $\left(\|\|\vec{x}\|\| = \sqrt{\vec{x} \cdot \vec{x}}\right)$.

B.3 Software

MATLAB	A numerical computing environment. Used for testing and simulations.
CTUstyle	A novel plainTEX template for thesis at CTU. Used to typeset this thesis. Created by Petr Olšák[1]).
PGF/Ti*k*Z	PGF is a TEX macro package for generating graphics with a user-friedly syntax layer called Ti*k*Z[2]). Used to program all of the graphics in this thesis (excluding photos). Created by Till Tantau.

[1]) http://petr.olsak.net/ctustyle.html
[2]) http://sourceforge.net/projects/pgf/

Appendix C
CD Content

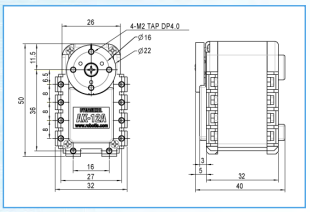
The enclosed CD contains an electronic copy of this thesis in it's root directory. It also contains the source files in a following directory structure

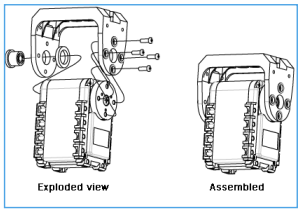
- /thesis — The source plainTEX files used to typeset this thesis.
- /robot — The source C++ files of the hexapod control program.
- /external — The source files of an external Dynamixel library.
- /matlab — The MATLAB scripts used to verify and simulate the gait behavior.

Appendix D
Hardware Datasheets

D.1 ROBOTIS Dynamixel Servos

Dynamixel
AX Series





AX Series Dyanmixels & Brackets
(Click to View Full Line)

Model	AX-12A (Visit Product Page)	AX-12W (Visit Product Page)	AX-18A (Visit Product Page)
Stall Torque @ Max Voltage	1.5N.m (16.5 kg-cm)	0.2N.m (2.0 kg-cm)	1.8N.m (18 kg-cm)
Speed (RPM)	59	470	97
Nominal Operating Voltage	12v	12v	12v
Stall Current Draw	1.5A	1.4A	2.2A
Dimensions	32x50x40 mm	32x50x40 mm	32x50x40 mm
Weight	54.6g	52.9g	54.5g
Resolution	0.29°	0.29°	0.29°
Operating Angle	300	300	300
Gear Reduction	254 : 1	32 : 1	254 : 1
Geartrain Material	Eng. Plastic	Eng. Plastic	Eng. Plastic
Onboard CPU	ATMega8 (ATMEGA8-16AU@16MHZ, 8 Bit)	ATMega8 (ATMEGA8-16AU@16MHZ, 8 Bit)	ATMega8 (ATMEGA8-16AU@16MHZ, 8 Bit)
Position Sensor	Potentiometer	Potentiometer	Potentiometer
Com Protocol	TTL	TTL	TTL
Com Speed	1mbps	1mbps	1mbps
Compliance/PID	Compliance	Compliance	Compliance
Dimensional Drawing	PDF	PDF	PDF

http://www.trossenrobotics.com

Appendix E
Equations

E.1 Direct Kinematics

We can create transformation matrices from a Denavit-Hartenberg notation [26] of a hexapod leg kinematics in Table 4.2. The transformation matrix $A^{4\times4}$ is in homogeneous coordinates, so it contains both rotation and translation.

The transformation matrix between two adjacent coordinate systems O_i and O_{i-1} is

$$A_i^{i-1} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{1}$$

The transformation matrices from the coxa joint (O_C) through the femur (O_F) and tibia joints (O_T) to the foot (O_W) looks as follows for our parameters of D-H notation.

$$A_F^C = \begin{bmatrix} \cos\theta'_C & 0 & \sin\theta'_C & a_C\cos\theta'_C \\ \sin\theta'_C & 0 & -\cos\theta'_C & a_C\sin\theta'_C \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$A_T^F = \begin{bmatrix} \cos\theta'_F & -\sin\theta'_F & 0 & a_F\cos\theta'_F \\ \sin\theta'_F & \cos\theta'_F & 0 & a_F\sin\theta'_F \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

$$A_W^T = \begin{bmatrix} \cos\theta'_T & -\sin\theta'_T & 0 & a_T\cos\theta'_T \\ \sin\theta'_T & \cos\theta'_T & 0 & a_T\sin\theta'_T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4}$$

where $\theta'_i = \theta_i + \theta_i^{off}$. The a_i actually stands for an i-th link length here.

The transformation matrix $A_W^C = A_F^C A_T^F A_W^T$ represents the mapping between the coxa joint coordinate system and the foot coordinate system. The foot coordinates lies in the origin of the coordinate system O_W. To get this coordinates in the coordinate system O_C (which is fixed with the hexapod body), we use the transformation matrix A_W^C.

$$A_W^C \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta'_C(a_C + a_T\cos(\theta'_F + \theta'_T) + a_F\cos\theta'_F) \\ \sin\theta'_C(a_C + a_T\cos(\theta'_F + \theta'_T) + a_F\cos\theta'_F) \\ a_T\sin(\theta'_F + \theta'_T) + a_F\sin\theta'_F \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{5}$$

The foot has coordinates $[x, y, z]^T$ in the coordinate system O_C (which lies in the coxa rotation axis) following the equation (5).

E.2 Inverse Kinematics

Given the foot coordinates $[x, y, z]$, the coxa joint angle θ_C can be computed directly from the x and y coordinates

$$\theta_C = \begin{cases} \text{atan2}(x, y) - \theta_C^{off} & \text{if } x \geq 0; \\ \text{atan2}(-x, -y) - \theta_C^{off} & \text{if } x < 0. \end{cases} \tag{6}$$

Next, we will simplify the notation by introducing the coordinates $[x^F, y^F, z^F]$ which are the foot coordinates expressed in the coordinate system O_F (see Appendix E.1).

$$\begin{aligned} x^F &= \sqrt{x^2 + y^2} - a_C \\ y^F &= z \\ z^F &= 0 \end{aligned} \tag{7}$$

We can also express the length of the leg from the femur joint (O_F) to the foot (O_W) as

$$d_{FW} = \sqrt{(x^F)^2 + (y^F)^2}.$$

According to the law of cosines and the angle above horizon, we get the femur and tibia angles θ_F, θ_T

$$\theta_F = \arccos\left(\frac{{a_F}^2 - {a_T}^2 + {d_{FW}}^2}{2a_F d_{FW}}\right) - \text{atan2}\left(y^F, x^F\right) - \theta_F^{off} \tag{8}$$

$$\theta_T = \pi - \arccos\left(\frac{{a_F}^2 + {a_T}^2 - {d_{FW}}^2}{2a_F a_T}\right) - \theta_T^{off}, \tag{9}$$

where a_i is actually the i-th link length and θ_i^{off} is the i-th joint offset (see Table 4.2).

This equations works for cordinates $[x, y, z]$ in the positive operating radius of a leg (see Figure 3.2). If we wanted to allow also the negative radius, a more complex solution especially of the coxa joint θ_C would have had to be done.

E.3 Linear Regression

Given a set of n coordinates (x_i, y_i, z_i) in 3D space, we can find a plane $z = ax + by + c$ in which the squared distance between each coordinate and the plane (in the z-axis) is minimized.

The sum can be written as $S = \sum_{i=1}^{n}[(ax_i + by_i + c) - z_i]^2$ and it's a nonnegative function of three variables $S(a, b, c)$. It's graph is a parabola in each variable and the minimum can be found using partial derivations

$$\begin{aligned} \frac{\partial S}{\partial a} &= 0 = 2\sum_{i=1}^{n}[(ax_i + by_i + c) - z_i]x_i \\ \frac{\partial S}{\partial b} &= 0 = 2\sum_{i=1}^{n}[(ax_i + by_i + c) - z_i]y_i \\ \frac{\partial S}{\partial c} &= 0 = 2\sum_{i=1}^{n}[(ax_i + by_i + c) - z_i]. \end{aligned} \tag{10}$$

These equation can be rewritten as

$$\begin{aligned}\sum_{i=1}^{n}(ax_i^2 + bx_iy_i + cx_i) &= \sum_{i=1}^{n} z_ix_i \\ \sum_{i=1}^{n}(ax_iy_i + by_i^2 + cy_i) &= \sum_{i=1}^{n} z_iy_i \\ \sum_{i=1}^{n}(ax_i \quad + by_i + c \quad) &= \sum_{i=1}^{n} z_i \end{aligned} \tag{11}$$

and in a matrix notation, we get

$$\begin{bmatrix} \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_iy_i & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_iy_i & \sum_{i=1}^n y_i^2 & \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n y_i & n \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n z_ix_i \\ \sum_{i=1}^n z_iy_i \\ \sum_{i=1}^n z_i \end{bmatrix}. \tag{12}$$

The parameters $[a, b, c]^T$ can be expressed by multiplying the equation by an inverse of the left matrix. Substituting the matrices, we get

$$X \begin{bmatrix} a \\ b \\ c \end{bmatrix} = Y \tag{13}$$

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = X^{-1}Y = \frac{\operatorname{adj} X}{\det X} Y. \tag{14}$$

Assuming that each sum is for $i = 1$ to n, we will simplify the notation using ($\sum \equiv \sum_{i=1}^n$). The determinant of X is therefore

$$\det X = n\sum x_i^2 \sum y_i^2 + 2\sum x_iy_i \sum y_i \sum x_i - (\sum x_i)^2 \sum y_i^2 - n(\sum x_iy_i)^2 - \sum x_i^2 (\sum y_i)^2.$$

The adjugate matrix ($\operatorname{adj} X$) can be expressed as

$$\operatorname{adj} X =$$

$$= \begin{bmatrix} n\sum y_i^2 - (\sum y_i)^2 & \sum x_i \sum y_i - n\sum x_iy_i & \sum x_iy_i \sum y_i - \sum x_i \sum y_i^2 \\ \sum x_i \sum y_i - n\sum x_iy_i & n\sum x_i^2 - (\sum x_i)^2 & \sum x_i \sum x_iy_i - \sum x_i^2 \sum y_i \\ \sum x_iy_i \sum y_i - \sum x_i \sum y_i^2 & \sum x_i \sum x_iy_i - \sum x_i^2 \sum y_i & \sum x_i^2 \sum y_i^2 - (\sum x_iy_i)^2 \end{bmatrix}.$$

The resulting parameters $[a, b, c]^T$ can be expressed by mutliplying the matrices above by following Equation (14).

E.4 Inverse Transformation

Given the transformation matrix A, we can separate it to the rotation and translation part.

$$A = \begin{bmatrix} R & R\vec{t} \\ 0\,0\,0 & 1 \end{bmatrix} = \begin{bmatrix} R & 0 \\ 0\,0\,0 & 1 \end{bmatrix} \begin{bmatrix} I & \vec{t} \\ 0\,0\,0 & 1 \end{bmatrix} \tag{15}$$

Following the identity $(XY)^{-1} = Y^{-1}X^{-1}$, the inverse of A can be performed by inverting the separated matrices.

$$\begin{aligned} A^{-1} &= \left(\begin{bmatrix} R & 0 \\ 0\,0\,0 & 1 \end{bmatrix} \begin{bmatrix} I & \vec{t} \\ 0\,0\,0 & 1 \end{bmatrix} \right)^{-1} \\ &= \begin{bmatrix} I & \vec{t} \\ 0\,0\,0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} R & 0 \\ 0\,0\,0 & 1 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} I & -\vec{t} \\ 0\,0\,0 & 1 \end{bmatrix} \begin{bmatrix} R^T & 0 \\ 0\,0\,0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} R^T & -\vec{t} \\ 0\,0\,0 & 1 \end{bmatrix} \end{aligned} \tag{16}$$